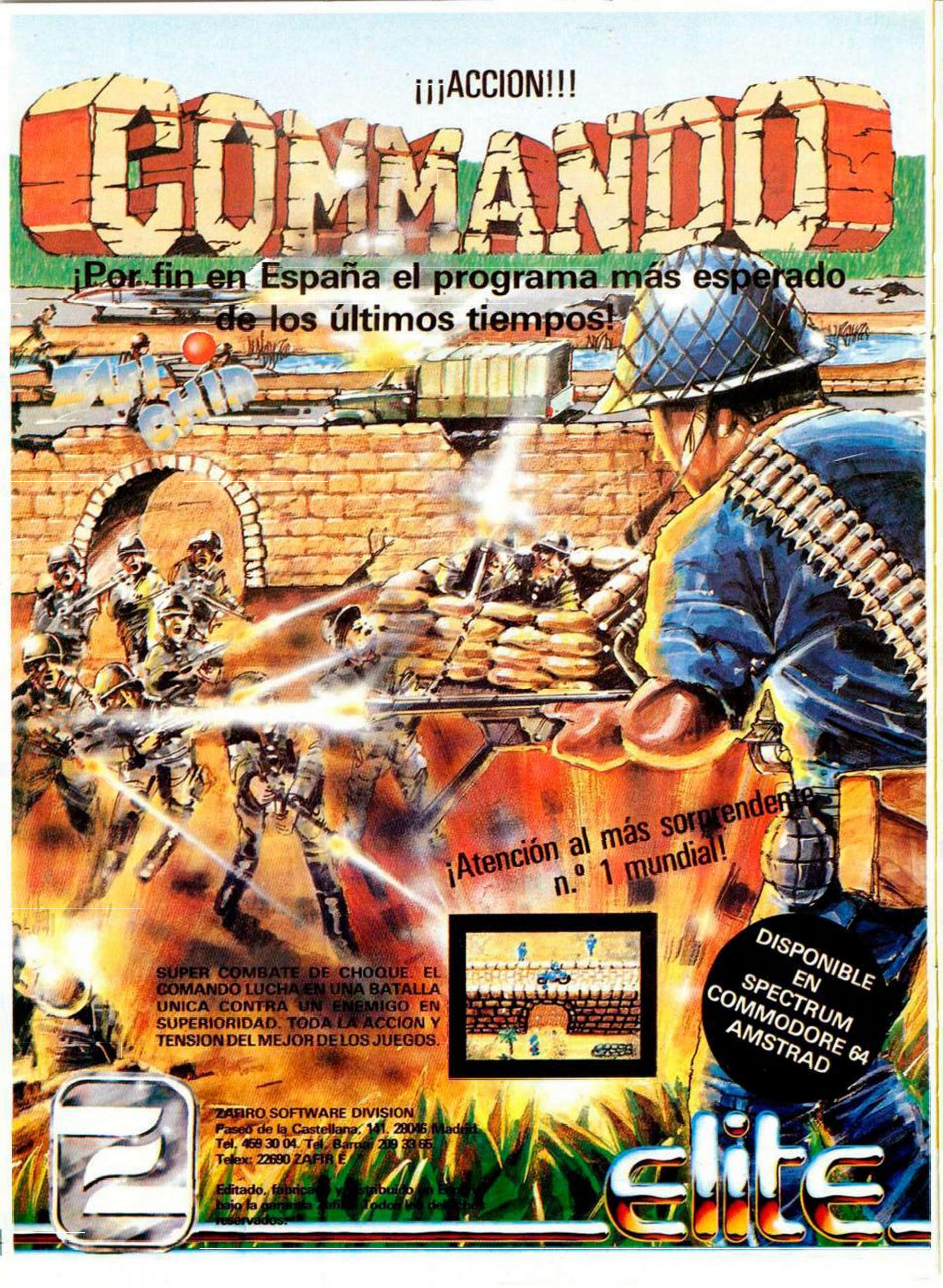


AÑO III - NUM. 80
MICROHOBBY
SEMANAL
REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR
AÑO III - N.º 80
135 PTS.
Canarias 140 ptas.
NUEVO
THE WAY OF THE TIGER
LA FUERZA DE LAS ARTES MARCIALES
ULTIMA HORA
PROYECTOS DE AMSTRAD SOBRE SINCLAIR!
INICIACION
COMO SE CONSTRUYE UN PROGRAMA COMERCIAL
TOP SECRET
PROTECCIONES ALEATORIAS
HOBBY PRESS


¡¡¡ACCION!!!
COMMANDO
¡Por fin en España el programa más esperado de los últimos tiempos!
¡Atención al más sorprendente n.º 1 mundial!
SUPER COMBATE DE CHOQUE. EL COMANDO LUCHA EN UNA BATALLA UNICA CONTRA UN ENEMIGO EN SUPERIORIDAD. TODA LA ACCION Y TENSION DEL MEJOR DE LOS JUEGOS.
DISPONIBLE EN SPECTRUM COMMODORE 64 AMSTRAD
ZAFIRO SOFTWARE DIVISION
Paseo de la Castellana, 141. 28046 Madrid
Tel. 459 30 04. Tel. Barna. 209 33 65.
Telex: 22690 ZAFIR E
elite

MICROHOBBY ESTA SEMANA

AÑO III. N.° 80. 27 de mayo al 2 de junio de 1986.
135 ptas. Canarias, Ceuta y Melilla: 130 ptas.
Sobretasa aérea para Canarias: 10 ptas.





Toda la técnica de las artes marciales en «The way of the tiger».

Director Editorial
José I. Gómez-Centurión

Director Ejecutivo
Domingo Gómez

Asesor Editorial
Gabriel Nieto

Redactora Jefe
Africa Pérez Tolosa

Diseño
Rosa María Capital

Redacción
Amalio Gómez, Pedro Pérez, Jesús Alonso

Secretaria Redacción
Carmen Santamaría

Colaboradores
Primitivo de Francisco, Rafael Prades, Miguel Sepúlveda, Sergio Martínez y J. M. Lazo

Corresponsal en Londres
Alan Heap

Fotografía
Carlos Candel

Portada
José María Ponce

Dibujos
Teo Mójica, F. L. Frontán, J. M. López Moreno, J. Igual, J. A. Calvo, Lóriga, J. Olivares

Edita
HOBBY PRESS, S. A.

Presidente
María Andrino

Consejero Delegado
José I. Gómez-Centurión

Jefe de Producción
Carlos Peropadre

Publicidad Barcelona
José Galán Cortés
Tels.: 303 10 22 - 313 71 76

Secretaría de Dirección
Pilar Aristizábal

Suscripciones
M.ª Rosa González
M.ª del Mar Calzada

Dto. Circulación
Paulino Blanco

Distribución
Coedis, S. A. Valencia, 245
Barcelona

Imprime
Rotedic, S. A. Ctra. de Irún, km 12,450 (MADRID)

Fotocomposición
Novocomp, S. A.
Nicolás Morales, 38-40

Fotomecánica
Grof
Ezequiel Solana, 16

Depósito Legal
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cía. Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1290 BUENOS AIRES (Argentina)

Solicitado control
OJD

MICROHOBBY NUMEROS ATRASADOS

Queremos poner en conocimiento de nuestros lectores que para conseguir números atrasados de MICROHOBBY SEMANAL, no tienen más que escribirnos indicándonos en sus cartas el número deseado y la forma de pago elegida de entre las tres modalidades que explicamos a continuación.
Una vez tramitado esto, recibirá en su casa el número solicitado al precio de 95 ptas. + 6 de IVA hasta el n.° 36, a 125 ptas. + 8 de IVA hasta el n.° 60 y a 135 ptas. desde el n.° 60 en adelante.



FORMAS DE PAGO

- Enviando talón bancario nominativo a Hobby Press, S. A. al apartado de Correos 54062 de Madrid.
- Mediante Giro Postal, indicando número y fecha del mismo.
- Con Tarjeta de Crédito (VISA o MASTER CHARGE), haciendo constar su número y fecha de caducidad.

ULTIMA HORA

LOS NUEVOS PROYECTOS DE AMSTRAD SOBRE SINCLAIR


Sir Clive Sinclair

Como ya sabéis por anteriores noticias de esta sección, Amstrad adquirió el pasado mes los derechos de distribución y comercialización de los productos Sinclair. Sin embargo, Sinclair no se ha dejado morir, trabajando intensamente en los nuevos proyectos: «Pandora» y «Enigma».

Amstrad, por su parte, parece que tiene intención de enfocar más su producción hacia la línea de «bussines machine» (negocios y utilidades), mientras que el Spectrum va a ver fortalecida su faceta lúdica.

Sin embargo, la mayor expectación la está acaparando en este momento el futuro de los ordenadores que ya existen (no los que están en desarrollo). Amstrad sacará en breve, probablemente para este verano, un nuevo SuperSpectrum que potenciará más, si cabe, las posibilidades de esta máquina.

En principio, el nuevo ordenador del que se desconoce el nombre, va a ser plenamente compatible con todos los modelos de Spectrum actuales aunque la capacidad de memoria será de 128 K.

Uno de los detalles más importantes que lo distinguirán de modelos anteriores será un cassette incorporado, parecido al que lleva el 464 de Amstrad.

El otro nuevo detalle es el interface de Joystick incorcorado, con lo que quedarán resueltas de una vez por todas las dudas sobre qué Interface comprar, debido a la gran diversificación de protocolos existentes. En este aspecto sólo queda la duda de por qué sistema se decantarán los fabricantes de Amstrad (todos los indicios apuntan a la norma Kempston).

Todos los componentes y planos están ya en Taiwán donde se va a proceder al montaje y producción de la máquina en cuestión. Esto permitirá un gran abaratamiento de los costos, que de momento ya se anuncian 50 libras por debajo del precio actual del Spectrum 128 K.

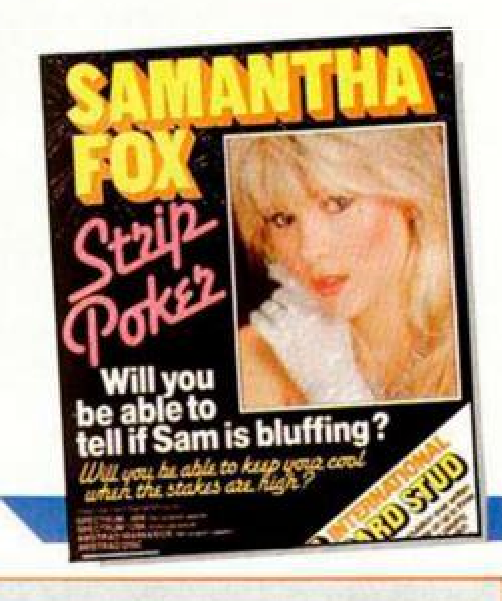
Sin embargo, lo que no está nada claro es el futuro del QL. Amstrad opina que esta máquina es, hasta cierto punto, competitiva con sus modelos 6128 y 8256. Por este motivo se aferra fuertemente a los derechos de producción que posee, más que nada para paralizar su difusión. Un consorcio de distribuidores encabezado por Tom Petty (diseñador del sistema operativo del QL) está intentando negociar los derechos de producción del QL. Independientemente de si lo consiguen o no, se espera que lancen al mercado un nuevo ordenador muy parecido al QL y del que hablaremos en próximas semanas: el QLT. Este será totalmente compatible con todo el software y hardware del QL y no va a tener nada que ver con Amstrad. Seguiremos informando.

EROTISMO EN EL ORDENADOR

Presentado por Martech nos llega desde Gran Bretaña un curioso y peculiar programa que responde al llamativo título de Samantha Fox, (La zorra Samantha).

El jueguecito en cuestión consiste en un Strip Póker en el cual la bellísima y erótica Samantha, una famosa actriz inglesa, nos va mostrando sus encantos personales (hasta los más íntimos) a medida que vayamos ganándole las diferentes manos de póker.

Este programa evidentemente tiene un encanto «muy especial» y de entre sus características cabe destacar muy notablemente los aspectos gráficos del mismo.



AQUI LONDRES

Amstrad está acreditando en los centros comerciales el nuevo juego de la compañía Ocean, **Batman** para el Spectrum 48 K y Amstrad ya que la versión para Commodore 64 no se podrá adquirir antes del verano.

El juego está basado en el personaje de cómic «el hombre murciélago» que ha divertido a la gente a lo largo de 50 años en tebeos y televisión. La misión de este programa consiste en salvar a su amigo Robin, quien ha sido secuestrado por los antiguos enemigos de Batman.

La primera simulación submarina acaba de ser lanzada por U.S. Gold bajo licencia de una compañía americana llamada **Mikro-Prose Software.** El juego se llama **«Slent Service»** (servicio silencioso) y ha sido escrito por el autor altamente reconocido de «F-15 Strike Eagle». De momento, se puede conseguir para el Commodore 64, pero las versiones para Spectrum y Amstrad serán adquiribles próximamente.

Las especulaciones que se vienen produciendo sobre la compra de los derechos exclusivos del QL ya han dejado de serlo. Parece ser que la oferta hecha por un consorcio de compañías que han venido abasteciendo a este ordenador no han conseguido comprar los derechos de **Amstrad** quien pretende aferrarse al QL para frenar su producción, ya que según piensan, este ordenador interfiere en las ventas de Amstrad.



De nuestro corresponsal en Londres ALAN HEAP

ENTREGA DE PREMIOS A LOS GANADORES DEL CONCURSO MASTERMIND

La pasada semana tuvo lugar en la central de Micro World, el acto de entrega de premios a los afortunados y brillantes ganadores del legendario Concurso Mastermind, que consistían, el 1.° en un viaje a Londres, el 2.° un monitor en color, el 3.° un Spectrum 128 K y para los 7 restantes, un lote de programas.

Este concurso, de carácter nacional, ha supuesto un gran esfuerzo por parte de todos, tanto de participantes como de organizadores, quienes han tenido que sufrir los inconvenientes que supone el organizar un concurso de este nivel en el que la afluencia de participación ha sido más que notable y en el que la propia dinámica de eliminatorias exigía un prolongado período de tiempo.

Sin embargo, tras más de un año de duración, el Concurso Mastermind ha llegado a su término y los ganadores han recibido un merecido premio a su ardua y paciente colaboración.



Pero el concurso no quería despedirse sin llevar a cabo una última anécdota. El primer clasificado, D. Fernando Rada (premiado con un viaje a Londres para dos personas) y D. Carlos Granados, clasificado en octava posición, son dos de los creadores del fenomenal programa Sir Fred.

Esto puede dar una idea del elevado nivel de dificultad que ha adquirido este concurso, pues incluso el brillante programador de todo un Sir Fred, ha quedado relegado a una honrosa octava posición. Pero también es verdad que cuando este concurso comenzó todos éramos un poco más jóvenes e inexpertos.

CLASIFICACION	SEMANAS PERM.	TENDENCIA	20 +	SPECTRUM	AMSTRAD	COMMODORE	MSX
1	2	–	**GREEN BERET.** Imagine	●			
2	3	–	**MOVIE.** Imagine	●			
3	2	↑	**THE WAY OF THE TIGER.** Gremlin Grafics	●	●	●	●
4	3	↓	**RAMBO.** Ocean	●	●	●	
5	3	↓	**TURBO ESPRIT.** Durell	●			
6	3	↑	**CAMELOT WARRIORS.** Dinamic	●			
7	2	↓	**PING PONG.** Imagine	●	●		●
8	2	↑	**COMANDO.** (Elite)				
9	2	↓	**CYBERUN.** Ultimate	●			
10	3	–	**SABOTEUR.** Durell	●		●	
11	3	↑	**POLE POSITION.** Atari	●	●		
12	3	↑	**MATCH DAY.** Imagine	●	●		
13	3	↓	**YIE AR KUN-FU.** Imagine	●	●	●	●
14	3	↓	**SIR FRED.** Made in Spain	●	●		
15	2	↓	**THE SOLD A MILLION.** US Gold	●			
16	3	↓	**THE EXPLODING FIST.** Melbourne House	●	●	●	
17	3	↑	**MICKIE.** Imagine	●	●		
18	1	↑	**WINTER GAMES.** Epix				
19	2	↑	**THREE WEEKS IN PARADISE.** Mikro-Gen	●			
20	3	↓	**WORLD SERIES BASKETBALL.** Imagine	●			

Esta información ha sido elaborada con la colaboración de los centros de Microinformática de El Corte Inglés.



Compuesto por 33 países, excluida España

CREACION DEL PROGRAMA INTERGUBERNAMENTAL DE INFORMATICA

La Oficina Regional de Ciencia y Tecnología para América Latina y el Caribe, ha hecho llegar hasta nuestra redacción una muy interesante carta, en la que se nos comunica la intención de crear un Programa Intergubernamental de Informática cuyos objetivos estarán primordialmente destinados a fortalecer la cooperación internacional y los esfuerzos nacionales, encaminados a la formación de especialistas, la creación de infraestructuras, la definición de políticas nacionales y el mejor reconocimiento de la función de la informática.

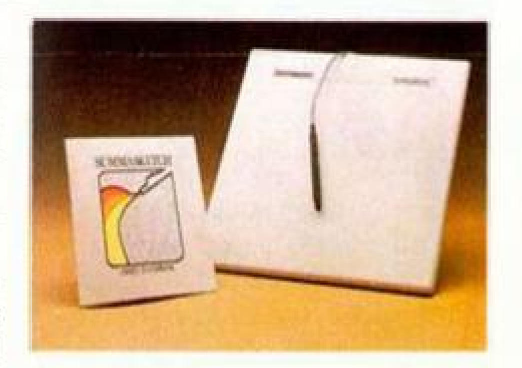
Esta organización estará compuesta por representantes de 33 Estados miembros, entre los que se incluyen países como Francia, República Federal de Alemania, Italia, Reino Unido, URSS, Argentina, Chile, Cuba e incluso forman parte de él estados como Gabón, Kenya, Nigeria, Sri Lanka o Bangladesh.

Sin embargo, por más que hemos buscado en esta extensa lista el nombre de España, no hemos conseguido encontrarlo por ningún sitio por la única razón de que no está inscrito en ella.

Desconocemos por completo las razones que han motivado la decisión de que España no pertenezca a este organismo internacional, el cual depende muy directamente de la Organización de Naciones Unidas (ONU) y simplemente nos limitaremos a hacernos eco de esta noticia que suponemos dejará desconcertado a más de un lector que a buen seguro se estará preguntando si España posee una política definida sobre los asuntos relacionados con la informática, sus especialistas y su infraestructura.

NUEVA GAMA DE PRODUCTOS PRESENTADA POR D.S.E.

La compañía distribuidora de sistemas electrónicos DSE., ha incorporado una nueva gama de productos para su comercialización en España.



Estas novedades son las siguientes:

SUMMASKETCH: una excelente herramienta para aquellos que deseen trabajar seriamente en el área de diseño gráfico. Esta tableta gráfica de alta calidad proporciona 100 líneas por pulgada de resolución y media pulgada de aproximación. Summasketch está confeccionado con tecnología de inducción magnética controlado por microprocesador y utiliza interface 25 Pin hembra y RS232C.



GEM COLLECTION: el paquete Gem consiste en un lote de software que, unido al hardware de Summagraphics, forman una excelente combinación para mejorar considerablemente la eficiencia gráfica del ordenador. En él se incluyen programas como Gem Write, el primer procesador de texto que trabaja de un modo totalmente gráfico; Gem Paint, un completo programa de diseño a todo color, y Gem Desktop, un gestor de entorno gráfico.

SUMMAMOUSE: este ratón electrónico sirve para obtener el mayor rendimiento del paquete Gem: un potente procesador de texto y un programa de diseño gráfico u otras utilidades, todo ello bajo el control de ratón. Interface RS232C.

Los precios de estos productos son de 43.000 ptas. para el ratón, 65.000 ptas. para el paquete GEM más el ratón, mientras que la tableta gráfica oscila entre las 95 y 232.000 ptas.



CHEQUEO DEL ESTADO DEL SPECTRUM

Con ayuda de esta pequeña rutina dentro de nuestros programas en Basic, podemos comprobar en cualquier momento cuánta memoria tenemos libre, si el modelo es de 16 ó 48 K, además de chequear cuánto ocupa el programa que tenemos en ese momento en el ordenador. También comprobaremos el tiempo que hace que empezamos a trabajar con el ordenador, lo que ocupan las variables y, si utilizamos el Microdrive, cuánta memoria nos ocupa éste, así como los gráficos definidos en el estado en el que se encuentran.

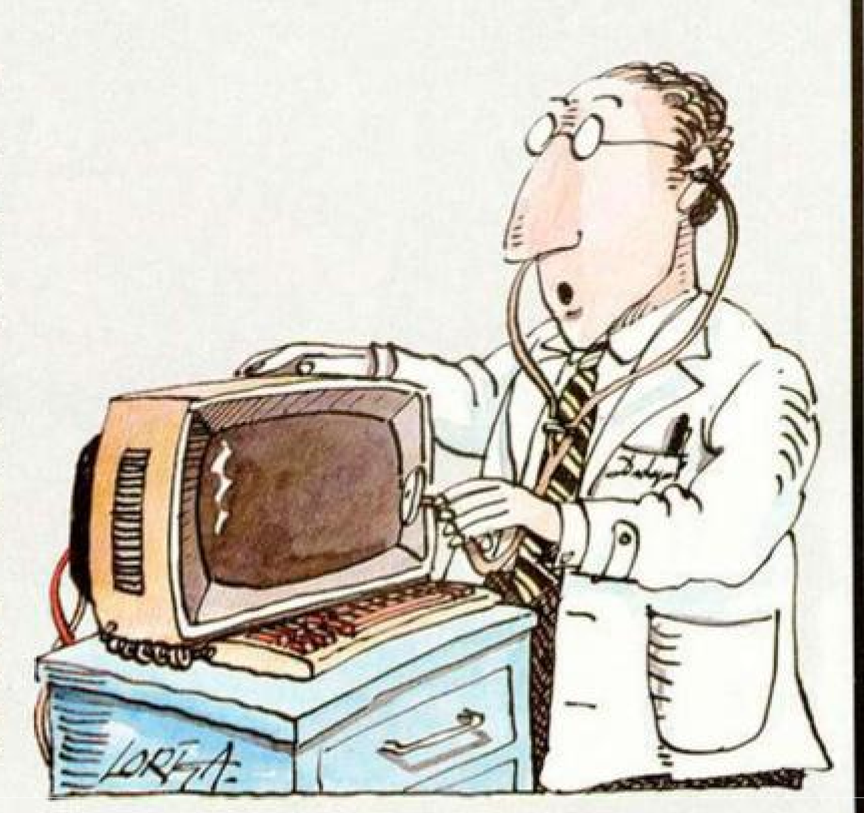
El funcionamiento es el siguiente:

1.— Tendremos que introducir en el programa la línea 10 aunque puede situarse en cualquier otra línea del programa siempre y cuando esté antes de la rutina.

2.— En la línea 10 utilizamos una definición de función con la que hallaremos más fácilmente las direcciones que vamos a utilizar.

3.— La línea 9999 calcula el tiempo y lo expresa en horas, minutos y segundos.

4.— El resto de las operaciones se realizan todas en la línea 9998.



```
  10 DEF FN p(a)=PEEK (a+1)*256+
PEEK a
9998 CLS : PRINT ';TAB 8;"MACHIN
E STATUS..";AT 4,2;"Total RAM
     :";(FN p(23732)-16383)/1024;
" K";AT 6,2;"Mapa Microdrive :";
FN p(23631)-23734;" bytes";AT 7,
2;AT 8,2;"Programa BASIC   :";FN
p(23627)-FN p(23635);" bytes";AT
 10,2;"Variables          :";FN p(2
3641)-FN p(23627)-1;" bytes";AT
12,2;"Espacio Libre   :";FN p(23
730)-FN p(23641);" bytes";AT 19,
2;"UDG: [redacted]"
9999 FOR x=1 TO 60: LET s=INT ((
FN p(23672)+65536*PEEK 23674)/50
): LET M=INT (S/60): LET H=INT (
M/60): PRINT AT 15,2;"Tiempo ";A
T 14,15;"I :";H;" horas ";AT 15
,15;"< :";INT (M-H*60);" minuto
s ";AT 16,15;"I :";INT (S-M*60);
"  segundos ": NEXT X
```

CONTRA EQUIVOCACIONES

Si probamos a introducir en el ordenador el siguiente POKE:

POKE 23614,97

y luego añadimos cualquier dato inexacto, es decir, que al pulsar ENTER nos aparezca un signo de interrogacion, veremos cómo el ordenador nos avisa con un NEW.

Con este truco de Casar Vega, conseguiremos pues, aprender a no equivocarnos al teclear sin cometer ningún error ya que utiliza la variable del sistema en el que se indica la dirección del retorno del error, cuyo nombre es la ERR SP.

¿Comprendido?

En este espacio también tienen cabida los trucos que nuestros lectores quieran proponer.

Para ello, no tienen más que enviarlos por correo a MICROHOBBY, Ctra. de Irún km 12,400 28049 Madrid.



DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA ERBE SOFTWARE
C/. STA. ENGRACIA, 17 - 28010 MADRID, TFNO. (91) 447 34 10
DELEGACION BARCELONA, AVDA. MISTRAL, Nº 10.
TFNO. (93) 432 07 31



ADEMAS DE
UN FABULOSO PROGRAMA DE FUTBOL, CONTIENE:

- UN ALBUM/POSTER EN EL QUE PODRAS SEGUIR LA COMPETICION PARTIDO A PARTIDO.
- UN POSTER CON LOS "MOMENTOS CUMBRE" DE LA HISTORIA DE LOS MUNDIALES A TODO COLOR.
- LOS BANDERINES AUTOADHESIVOS DE LOS PAISES PARTICIPANTES.
- UNA PEGATINA DE TELA CON EL EMBLEMA DE LOS MUNDIALES.

LAS BOLAS

TODAS LAS LETRAS MAYUSCULAS SUBRAYADAS DEBERAN TECLEARSE EN MODO GRAFICO.

Javier ROMERO

Spectrum 16 K

El más aguerrido ladrón del barrio ha conseguido adentrarse en el Edificio Maldito, un siniestro lugar que sólo tiene un atractivo: muchas bolsas de dinero escondidas estratégicamente.

Por esta razón, nuestro intrépido protagonista se ha arriesgado en esta acción sin conocer exactamente los peligros que le esperan. Focos de fuego y bolas asesinas que intentarán aplastarle, le esperan. ¿Conseguirá salir ileso y..., millonario?



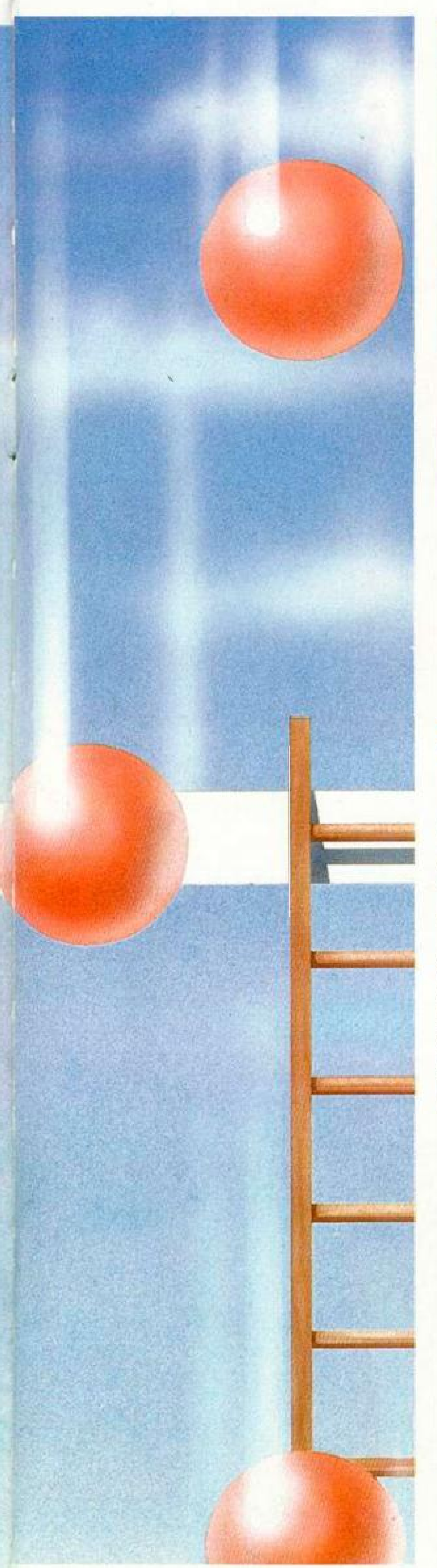


```
   2 GO SUB 5000
   3 POKE 23658,8
   5 PAPER 0: INK 7: BRIGHT 1: BORDER 1: CLS
  17 PAPER 0: INK 7: BRIGHT 1: BORDER 1: CLS
  20 LET AT=71: LET t=0: LET pant=1: LET pa=1: LET tes=0: LET vi=1
  30 LET dir=0: LET x=16: LET y=15
  50 REM ESCENARIO
 100 PRINT INK 4; PAPER 7;AT 2,0;"IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII"
 110 PRINT INK 3; PAPER 7;AT 6,0;"IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII"
 120 PRINT INK 1; PAPER 6;AT 10,0;"IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII"
 130 PRINT INK 5; PAPER 2;AT 14,0;"IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII"
 140 PRINT INK 2; PAPER 6;AT 18,0;"IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII"
 160 PRINT AT 2,5;"B";AT 3,5;"B";AT 2,12;"B";AT 3,12;"B";AT 2,22;"B";AT 3,22;"B"
 170 PRINT AT 6,9;"B";AT 7,9;"B";AT 6,18;"B";AT 7,18;"B";AT 6,26;"B";AT 7,26;"B";AT 6,29;"B";AT 7,29;"B"
 176 PRINT INK 5;AT 14,28;"KKJ"; INK 4;AT 2,13;"J";AT 2,18;"KJKK"; INK 1;AT 10,1;"K"
 177 PRINT INK 5;AT 14,10;"JK";AT 14,14;"J"
 178 PRINT INK 2;AT 18,29;"KJ"
 180 PRINT AT 10,4;"B";AT 11,4;"B";AT 10,14;"B";AT 11,14;"B";AT 10,22;"B";AT 11,22;"B"
 190 PRINT AT 14,5;"B";AT 15,5;"B";AT 14,21;"B";AT 15,21;"B";AT 14,27;"B";AT 15,27;"B";AT 19,24;"B"
 191 PRINT INK 1; PAPER 6; FLASH 1;AT 2,15;"C";AT 6,10;"C";AT 6,23;"C"
 192 PRINT INK 1; PAPER 6; FLASH 1;AT 10,2;"C";AT 10,10;"CCCC";AT 10,28;"C";AT 10,31;"C"
 193 PRINT INK 1; PAPER 6; FLASH 1;AT 14,15;"C";AT 14,7;"C";AT 18,8;"CCC";AT 18,25;"CCC"
```



```
 194 PRINT INK 6; PAPER 2; FLASH 1;AT 1,14;"G";AT 1,16;"G";AT 5,11;"G"
 195 PRINT INK 6; PAPER 2; FLASH 1;AT 5,24;"G";AT 9,3;"G";AT 9,30;"G";AT 13,31;"G";AT 17,8;"G"
 200 PRINT AT 21,0;"[redacted]"
 208 PRINT INVERSE 1;AT 21,0;"VIDAS: ";vi;AT 21,9;"TESOROS: ";tes;AT 21,21;"PANTALLA: ";pa
 209 LET a=1: LET b=-1
 210 LET c=1: LET d=7
 215 LET e=1: LET f=29
 216 GO SUB 221: GO TO 310
 219 REM SITUACION FLECHA
 221 LET p=INT (RND*5)
 222 IF p=0 THEN LET fle=0
 223 IF p=1 THEN LET fle=4
 224 IF p=2 THEN LET fle=8
 225 IF p=3 THEN LET fle=12
 226 IF p=4 THEN LET fle=16
 227 BEEP .02,10: BEEP .01,0
 228 RETURN
 229 REM BUCLE PRINCIPAL
 310 LET dir=dir+1: LET b=b+1: LET d=d+1: LET f=f-1
 315 PRINT INK 4;AT a,b;"A"; INK 2;AT c,d;"A"; INK 3;AT e,f;"A"
 316 PRINT AT fle,dir;"H"
 317 IF fle=x AND dir=y THEN GO SUB 4000
 328 PRINT AT x,y;"E";AT x+1,y;"F"
 329 REM BOLAS CON POS. a,b
 330 IF x+1=a AND y=b THEN GO SUB 4000
 340 IF b=5 OR b=9 OR b=14 OR b=21 THEN BEEP .01,5: LET a=a+4: PRINT INK 4;AT a-4,b;" ";AT a,b;"D"
 380 IF b=31 THEN PRINT INK 4;AT a,b;" ": LET a=1: LET b=-1
 389 REM BOLAS CON POS. c,d
 400 IF x+1=c AND y=d THEN GO SUB 4000
 410 IF d=12 OR d=18 OR d=22 OR d=27 THEN BEEP .01,8: LET c=c+4: PRINT INK 2;AT c-4,d;" ";AT c,d;"D"
 430 IF d=31 THEN PRINT AT c,d;" ": LET c=1: LET d=7
 455 REM BOLAS CON POS. e,f
 456 IF x+1=e AND y=f THEN GO SUB 4000
```





```
 460 IF f=22 OR f=18 OR f=14 OR f=5 THEN BEEP .01,-5: LET e=e+4: PRINT INK 3;AT e-4,f;" ";AT e,f;"D"
 492 IF f=0 THEN LET e=1: LET f=28: PRINT AT 17,0;" "
 500 REM INKEY$
 510 LET i$=INKEY$
 520 IF i$="Q" THEN IF x>0 THEN IF ATTR (x-2,y)=at THEN PRINT AT x,y;" ";AT x+1,y;" ": LET x=x-4
 530 IF i$="A" THEN IF x<21 THEN IF ATTR (x+2,y)=at THEN PRINT AT x,y;" ";AT x+1,y;" ": LET x=x+4
 540 IF i$="O" THEN IF y>0 THEN LET y=y-1: PRINT AT x,y+1;" ";AT x+1,y+1;" "
 550 IF i$="P" THEN IF y<31 THEN LET y=y+1: PRINT AT x,y-1;" ";AT x+1,y-1;" "
 560 IF x+1=a AND y=b OR x+1=c AND y=d OR x+1=e AND y=f OR fle=x AND dir=y THEN GO SUB 4000
 800 IF ATTR (x+2,y)=241 THEN GO SUB 4000
 810 IF dir>=31 THEN PRINT AT fle,31;" ": LET dir=0: GO SUB 221
 820 PRINT AT x,y;"E";AT x+1,y;"E"
 830 IF ATTR (x+1,y+1)=214 THEN GO SUB 3000
 840 IF ATTR (x+1,y-1)=214 THEN GO SUB 3050
 850 IF t=8 THEN GO SUB 8000
 860 IF x=20 THEN GO TO 6000
 900 PRINT AT fle,dir;" ";AT a,b;" ";AT c,d;" ";AT e,f;" "
 910 GO TO 310
2999 REM RECOGIDA DE OBJETOS
3000 PRINT AT x+1,y+1;" ": LET tes=tes+1: LET t=t+1
3005 PRINT AT 21,17; INVERSE 1;tes
3010 BEEP .01,13: BEEP .01,1
3020 RETURN
3050 PRINT AT x+1,y-1;" ": LET tes=tes+1: LET t=t+1
3055 PRINT AT 21,17; INVERSE 1;tes
3060 BEEP .01,13: BEEP .01,1
3070 RETURN
4000 REM MUERTO
4010 FOR k=0 TO 50: BEEP .01,INT (RND*40): PRINT INK INT (RND*7);AT x,y;"E";AT x+1,y;"E": NEXT k: PRINT AT x,y;" ";AT x+1,y;" "
4020 LET x=16: LET y=15: LET vi=vi-1
4022 IF vi=-1 THEN GO TO 7000
4025 PRINT AT 21,6; INVERSE 1;vi
4030 RETURN
4999 REM G.D.U.
5000 FOR f=0 TO 87
5010 READ s
5020 POKE USR "a"+f,s
5030 NEXT f
5040 DATA 60,126,255,255,255,255,126,60
5050 DATA 195,255,255,195,195,255,255,195
5060 DATA 170,85,170,85,170,85,170,85
5070 DATA 0,0,60,126,255,255,126,60
5080 DATA 60,60,126,60,60,24,60,126
5090 DATA 126,189,189,165,36,36,36,102
5100 DATA 36,66,189,32,60,133,126,36
5110 DATA 0,0,2,255,2,0,0,0
5520 DATA 247,247,247,0,127,127,127,0
5530 DATA 255,247,119,35,3,1,0,0
5540 DATA 255,255,254,190,156,12,8,0
5550 RETURN
5999 REM OTRA PANTALLA
6000 IF pant=1 THEN LET pant=2: LET pa=pa+1: LET at=112: INK 0: PAPER 6: BORDER 2: CLS : GO TO 30
6010 IF pant=2 THEN LET pant=1: LET pa=pa+1: LET at=71: INK 7: PAPER 0: BORDER 1: CLS : GO TO 30
6999 REM FINAL DE JUEGO
7080 PRINT INVERSE 1;AT 9,3;"                          "
7090 PRINT INVERSE 1;AT 11,3;"                          "
7100 PRINT INVERSE 1;AT 10,3;" QUIERES JUGAR DE NUEVO?   "
7110 LET i$=INKEY$
7120 IF i$="s" OR i$="S" THEN CLS : GO TO 17
7130 IF i$="n" OR i$="N" THEN STOP
7140 GO TO 7110
7999 REM IMPRIME SALIDA
8000 LET t=0: PRINT AT 18,24;"B";AT 20,22; FLASH 1;"SALIDA": RETURN
```

INTERCAMBIO

Miguel SEPULVEDA

Spectrum 16 K

Este juego consiste en pasar las fichas de color negro a donde están las rojas y viceversa, teniendo en cuenta que las negras avanzan de izquierda a derecha y las rojas al contrario. Ninguna puede retroceder y no se puede poner una ficha donde hay otra. Para controlarlas contamos con dos teclas: 8, para borrar datos erróneos y 0, para comenzar la partida.

```
   1 REM M. Sepulveda *INTERCAMBIO*
   5 DEF FN a()=(ATTR (3,x*4))-32
   9 GO SUB 1000
  10 GO SUB 1050: GO TO 100
  11 LET r=0: PRINT FLASH 1;AT 21,23;"MUEVE"
  12 PAUSE 0: LET x$=INKEY$: IF LEN x$>1 OR x$<"0" OR x$>"8" THEN GO TO 12
  13 IF x$="8" THEN PRINT AT l,c;" - ": GO TO 11
  14 IF x$="0" THEN GO SUB 1050: GO TO 11
  15 LET x=CODE x$-48: IF r=0 THEN LET y=x: LET b=FN a(): LET r=r+1: PRINT AT l,c;x$: GO TO 12
  16 LET z=x: LET d=FN a(): PRINT AT l,c+2;x$
  20 IF b=7 OR d<>7 THEN LET r=0: RETURN
  21 IF b=2 THEN GO TO 24
  22 IF z>y+2 OR z<y THEN LET r=0
  23 RETURN
  24 IF z<y-2 OR z>y THEN LET r=0
  25 RETURN
 100 GO SUB 11
 110 IF r=0 THEN PRINT AT 21,0;"MOVIMIENTO INCORRECTO.";AT l,c;" - ": GO TO 100
 120 PRINT AT 21,0;"                                "
 130 PRINT INK 7;AT 3,y*4-1;a$;AT 4,y*4-1;b$; INK b;AT 3,z*4-1;a$;AT 4,z*4-1;b$
 140 LET b=0: LET d=b
 150 FOR n=1 TO 3: LET x=n: LET b=b+FN a(): LET x=n+4: LET d=d+FN a(): NEXT n
 160 LET l=l+1: IF l>19 THEN LET l=8: LET c=c+5
 170 IF b<>6 OR d<>0 THEN GO TO 100
 180 FOR n=1 TO 50: BORDER 6: BORDER 2: BORDER 6: BEEP .2,10: BORDER 3: BORDER 7: BORDER 5: BEEP .1,30: BORDER 0: BORDER 5: BORDER 4: BEEP .3,20: NEXT n
 190 INPUT AT 0,0;"Quieres volver a intentarlo ";LINE c$: IF c$="s" THEN GO TO 10
 200 STOP
1010 FOR n=USR "a" TO USR "a"+31: READ a: POKE n,a: NEXT n: RETURN
1020 DATA 3,15,31,63,127,127,255,255, 192,240,248,252,254,254,255,255
1030 DATA 255,255,127,127,63,31,15,3, 255,255,254,254,252,248,240,192
1050 BORDER 0: PAPER 4: INK 7: CLS
1060 LET a$="AB": LET b$="CD": PRINT AT 0,10;"INTERCAMBIO";AT 3,15;a$;AT 4,15;b$
1070 FOR n=3 TO 27 STEP 4: PRINT AT 2,n;INT (n/4+1): NEXT n
1080 FOR n=3 TO 11 STEP 4: PRINT INK 0;AT 3,n;a$;AT 4,n;b$; INK 2;AT 3,n+16;a$;AT 4,n+16;b$: NEXT n
1090 PRINT AT 6,10;"MOVIMIENTOS"
1100 FOR n=8 TO 19: PRINT AT n,13;"-";AT n,18;"-": NEXT n
1110 LET l=8: LET c=12: RETURN
```

THE WAY OF THE TIGER • Deportivo • Gremlin Graphics

LAS PRUEBAS DEL NINJA

El Gran Maestro Naijishi ha dedicado gran parte de su vida y prácticamente la totalidad de la tuya, en hacer de ti un verdadero Ninja. Ahora te ha llegado el momento de demostrar que eres digno de recibir tan siquiera el nombre de 'alumno' del Maestro. Eso de llegar a Ninja, aún está por ver...

En los últimos meses estamos siendo invadidos por una avalancha de programas cuyo argumento se centra en alguna de las cientos de diferentes artes marciales existentes.

Sin embargo, debemos reconocer que la gran mayoría de estos programas han sido capaces de alcanzar un elevado nivel de calidad y se puede decir que con cualquiera de ellos la diversión está asegurada. De esta forma podríamos citar a programas tan famosos como The Way of the Exploding Fist (la primera gran obra en este tipo de juegos), Yie Ar Kun Fu, International Kárate..., los cuales han conseguido, cada uno con sus características y estilos particulares, resultados altamente positivos tanto desde el punto de vista del divertimento como del puramente estético.





Pero la producción en este campo es inagotable y parece que la fuente de la que manan las ideas para confeccionar estos programas es inagotable. Así, siempre surge un nuevo programa que es capaz de despertar el interés y la admiración de los adictos.

¿De qué estamos hablando? De The Way of the Tiger, por supuesto.

Y es que los señores de Gremlin Graphics se las han ingeniado para crear un programa de lo más divertido y emocionante, e incluso han conseguido que un tema tan tratado ya, vuelva a parecer algo original.

Esto lo han conseguido principalmente gracias al número tres: tres pruebas diferentes y triple scroll de







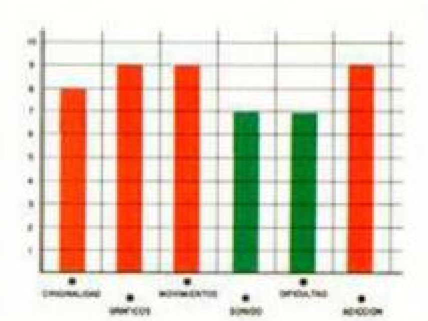
pantalla. Pero vayamos por partes.

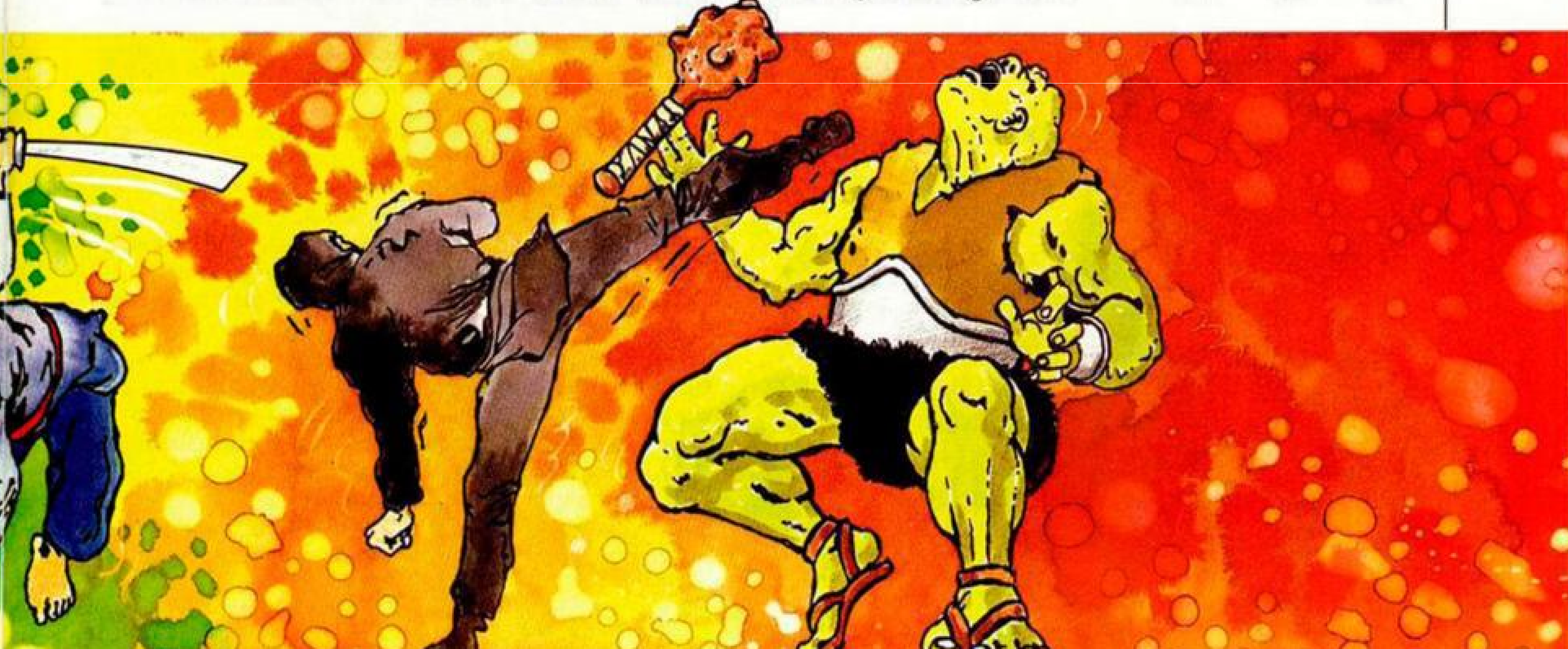
Tres pruebas diferentes: Unarmed Combat, Pole Fighting y Sword Fighting, o lo que es lo mismo, una primera fase de combates cuerpo a cuerpo, sin más armas que nuestros puños y más defensa que nuestra astucia y rapidez; una segunda de lucha con palos, en la que tendremos la oportunidad de combatir con los más fantasmagóricos rivales contando con el apoyo de un simple bastón, y por último, la gran lucha con espadas, la prueba que será el eslabón definitivo para alcanzar el anhelado título de Ninja.

Triple scroll: un sistema revolucionario de programación con el cual se consigue un excelente efecto visual debido a que al existir tres niveles diferentes en movimiento, la sensación de profundidad y realismo se hacen patentes durante todo el desarrollo del programa.

Pero todo esto podría ser incluso insuficiente si no se acompañara de unos excelentes movimientos, una estupenda presentación en pantalla y unos diseños diferentes. En una palabra, que a simple vista se nota que The Way of the Tiger no ha sido realizado de la noche a la mañana y que detrás de los casi 148 k de memoria que ocupa el programa en su totalidad (pues debemos aclarar que cada prueba debe ser cargada por separado) se encuentra una importante labor de programación.

¡Vamos, que es un juegazo!

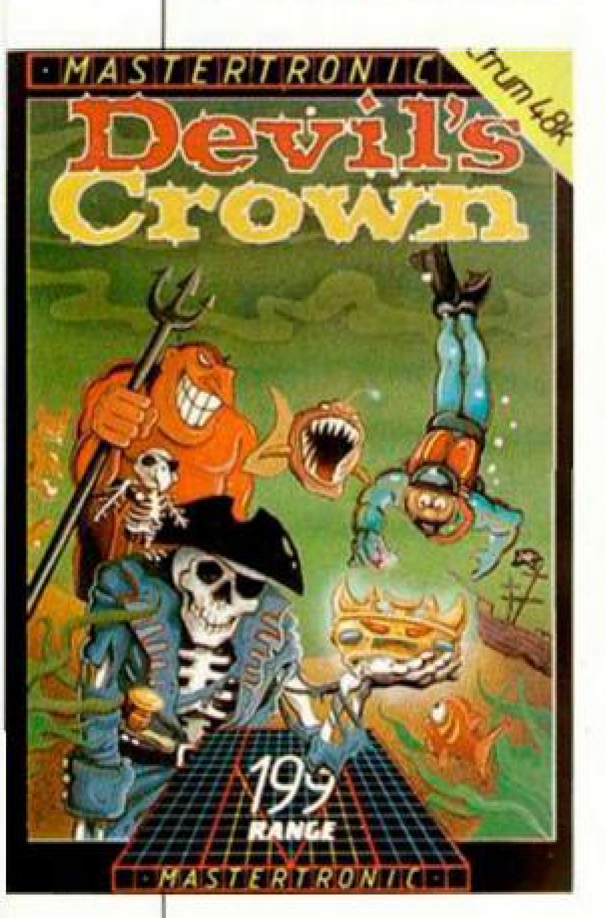




¡NUEVO!

THE DEVILS CROWN • Videoaventura • Mastertronics

EN LAS PROFUNDIDADES OCEANICAS



Mastertronics es una compañía británica de software que se ha hecho famosa por la comercialización de sus programas a un precio sorprendentemente más bajo de lo que viene siendo habitual en aquel mercado.

Si tenemos en cuenta que en Gran Bretaña un juego de los llamados de alta calidad viene a costar entre las 8 y 10 libras (1.700-2.000 ptas.) no es de extrañar que a este tipo de juegos, que evidentemente alcanzan el nivel de los anteriores y cuyo precio es de tan sólo 2 libras, es decir, algo menos de 450 ptas., reciban la calificación de software barato.

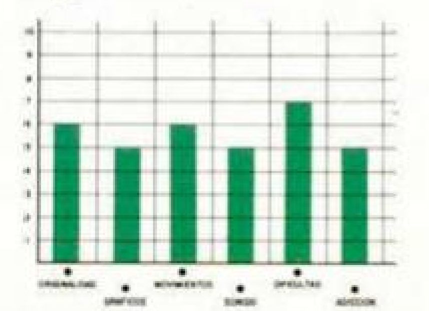
Sin embargo, a pesar de este precio tan reducido (algo casi inimaginable en nuestro país), estos juegos tampoco resultan malos del todo. Por supuesto que no se les puede exigir una calidad suprema, pero sí poseen una buena relación calidad/precio.

En el caso particular que hoy nos ocupa, The Devils Crown, se trata de una complicada videoaventura que ha nacido como versión de un buen programa que, con el mismo nombre, ya lleva algún tiempo funcionando en Commodore, sin embargo, poco tienen que ver el uno con el otro, tanto en los gráficos como en el argumento en sí de los mismos.

El juego va de piratas, de lo que se deduce fácilmente que el argumento consiste en buscar un tesoro. Este se encuentra oculto en el interior de una antigua nave, hundida en la profunda oscuridad del océano. Para llegar a conseguir nuestro objetivo deberemos recorrer las 40 habitaciones (camarotes para ser más exactos) en busca de los objetos necesarios para llegar a recobrar los maravillosos Siete Diamantes.

El buque se encuentra plagado de almas en pena y espíritus de bucaneros que, junto a las peligrosísimas pirañas, intentarán evitar que salgamos con vida o con las joyas.

Resumiendo: un programa del montón, más bien pobre de ideas y de gráficos, pero que por el precio que tiene no se le puede exigir mucho más.





SPECVENTURE • Arcade • Elisa

A LA BUSQUEDA DE LOS BYTES

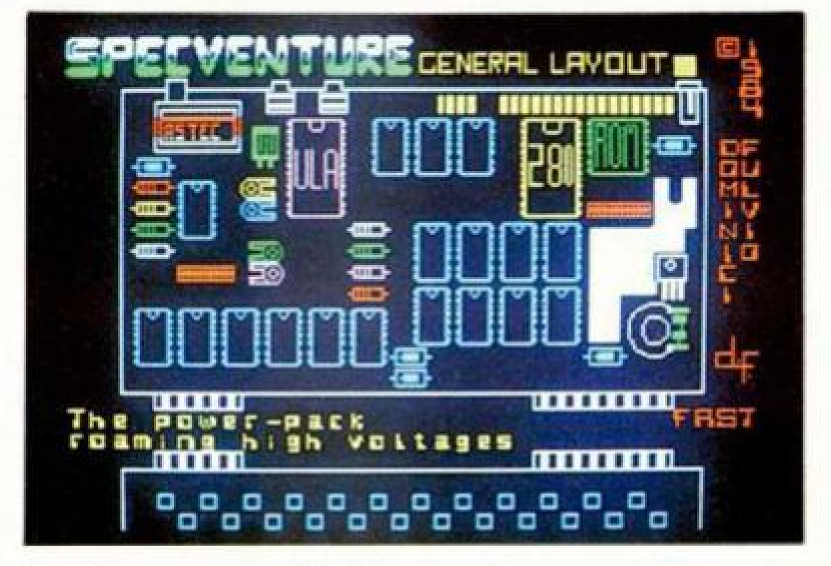
En Specventure representamos a un robot computerizado cuyo único afán y objetivo es el de recoger todos los bytes que han escapado del sistema operativo y volver a colocarlos en su lugar adecuado. Una vez conseguido esto, podremos salir del Spectrum a través de su altavoz.

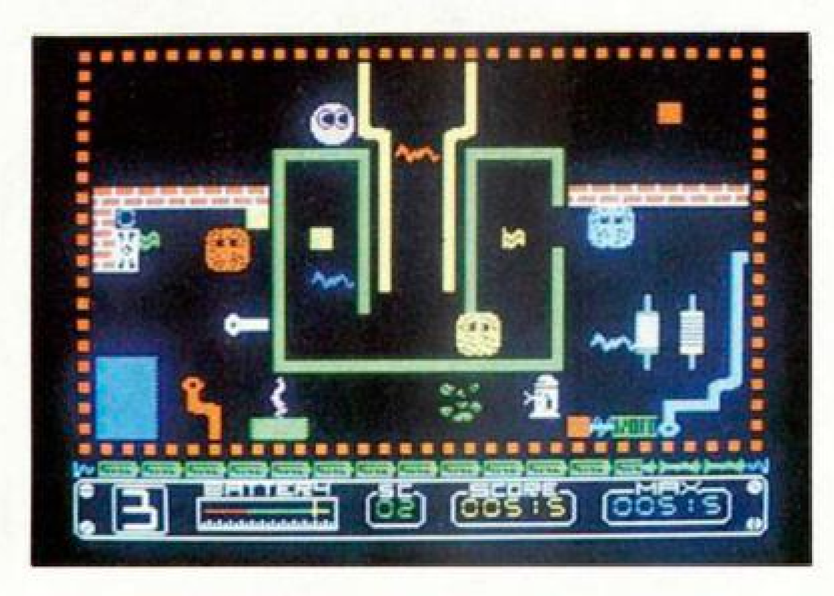
Como cabría esperar, los duendes que siempre habitan en el interior de los ordenadores van a intentar, con bastante éxito por su parte, el hacernos que las cosas nos resulten más complicadas, por lo que además de procurar no recibir ninguna descarga eléctrica de algún circuito mal conectado, tendremos también que eludir las continuas persecuciones de estos numerosos enemigos.

Afortunadamente contamos con un arma lanzadescargas que nos servirá de mucha ayuda en momentos decisivos y durante todo el transcurso de juego en general, pues aunque en algunas ocasiones podemos prescindir de ella, bien es verdad que nos facilitará la labor y hará que consigamos una mayor cantidad de puntos (que siempre vienen bien).

El juego tiene lugar en 30 diferentes escenarios, correspondiendo cada uno de ellos a una parte específica de los circuitos del Spectrum.

Una vez que hayamos recogido todos los bytes co-

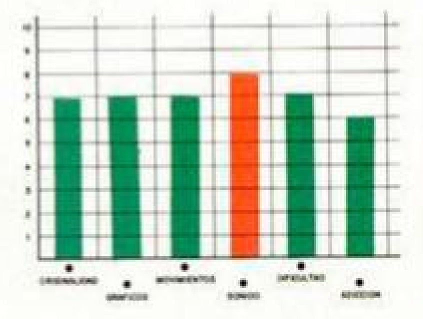




rrespondientes a una pantalla, podremos conectar el circuito de salida y pasar a la siguiente parte del sistema operativo del ordenador.

Gráficamente este Specventure no es demasiado llamativo, pero los diseños de las diferentes pantallas resultan lo suficientemente originales, simpáticos y decorativos como para decir que la calidad de diseño del programa es bastante aceptable.

Uno de los detalles que más llama la atención es el aspecto del sonido que acompaña al juego, el cual presenta la particularidad de que nos ofrece la posibilidad de elegir entre nueve sintonías diferentes que podremos seleccionar durante el transcurso del juego y que van desde Para Elisa, de Beethoven, hasta el tema de la Guerra de las Galaxias.



GERRY THE GERM • Arcade • Firebird

EL VIRUS



Gerry the Germ pertenece a un lote de programas que Firebird ha lanzado al mercado simultáneamente y a un precio bastante reducido. Este está compuesto por títulos como Costa Capers o el mismísimo Rasputín, y quizás de entre todos ellos este Gerry sea el que presenta una calidad ligeramente inferior.

El argumento del juego consiste en un simpático y travieso virus, el cual se ha propuesto adentrarse en un cuerpo humano, llegar hasta el mismísimo corazón y provocar allí un golpe definitivo: el colapso.

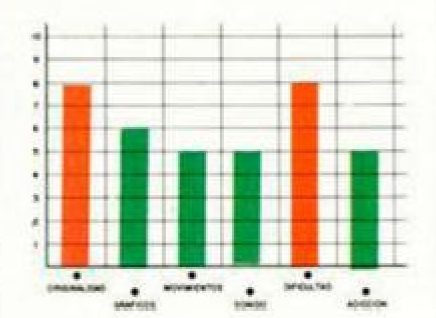
Para ello tendrá que ir recorriendo previamente algunos de los órganos y sistemas más importantes del cuerpo. La primera de las pruebas comenzará en los mismísimos pulmones, donde deberá recoger el oxígeno necesario que le permita llegar hasta los riñones. Si allí consigue comer las suficientes judías como para tener que ir a hacer una visita al Sr. Roca (ustedes ya me entienden), podrá adentrarse directamente en la vejiga urinaria propiamente dicha.



De allí deberá dirigirse al estómago, posteriormente al páncreas y así, una vez que haya acabado con todas las defensas de ese órgano, tendrá la oportunidad de realizar su ataque final *en el corazón.*

Como podréis comprobar, el argumento es de lo más original y divertido, pero hemos de reconocer que a la hora de la verdad, el juego resulta poco claro, algo confuso en lo que debemos o no debemos de hacer y, en definitiva, esta complicación le hace perder gran parte de su gracia, por lo que acaba siendo un juego algo pobre.





Por otra parte, los gráficos, aunque también muy originales por la forma tan personal y divertida que tienen de ver cada uno de los órganos del cuerpo, tampoco son la sensación de ser excesivamente buenos. Un programa original. Pero no sólo de originalidad viven los juegos.

Profesor particular

Arturo LOBO y J. J. LEON

PLANO AFIN 2

Esta es la segunda parte que os habíamos prometido publicar acerca del plano afín. La forma de utilizarla es haciendo un MERGE encima del programa PLANO AFIN 1, con lo que se os abrirá un nuevo abanico de posibilidades que pasamos a describir:

Teclear RUN y aparecerá un menú general con cuatro opciones, de las cuales la primera coincide con el programa 1 aunque con ciertas modificaciones. La más importante de ellas la tendréis a la vista: Es la posibilidad de guardar rectas en memoria. Cuando las ecuaciones de una recta estén en pantalla, pulsando una tecla del 1 al 9, la podréis mantener en una de las nueve memorias, lo que os servirá no sólo para tener acceso a ellas en cualquier momento sino también para utilizar muchas de las restantes opciones del programa. Además, se incluyen una serie de modificaciones que permiten trabajar con rectas horizontales y verticales, así como con aquéllas que pasen por el origen que en el programa 1 podían dar dificultades por el hecho de no admitir ciertas formas de la ecuación.

La segunda posibilidad del menú general, o sea paralelismo y perpendicularidad, nos permite hacer tres cosas: Hallar una paralela o una perpendicular a una recta en memoria, (dando un punto por el que tenga que pasar) y comprobar la relación entre dos rectas en memoria (si son paralelas o perpendiculares entre sí o no).

La tercera opción, cambios de base, pide dos vectores que guarda en memoria (y que deben ser linealmente independientes) y los considera una base, proporcionando las componentes de cualquier punto o vector en esta nueva base a partir de sus coordenadas actuales. Esta opción propia de espacios vectoriales se incluye dado el gran engorro que supone efectuar el cambio a base de cálculos.

Y por último, la cuarta opción, Distancias e intersecciones, ofrece también 3 posibilidades: Hallar la distancia entre dos puntos que se introduzcan, hallar la mínima distancia de un punto a una recta en memoria y hallar el punto de corte entre dos rectas (o sea, resolver el sistema que constituyen) aunque no está previsto resolver una recta horizontal con otra vertical por ser la solución obvia.

En resumen, esperamos que saquéis el máximo partido a este programa y os recordamos que posee un particular modo de introducir los datos numéricos (ya que trabaja con fracciones) que os explicábamos junto con «plano afín 1» y que se mantiene en la mezcla de los dos.

```
  10 GO TO 70
  20 CLS : PRINT AT 3,3; INVERSE
 1;"MENU GENERAL": RESTORE 20: F
OR i=1 TO 4: READ a$: PRINT AT 4
+3*i,6;i;")";a$: BEEP .5,3*i: NE
XT i
  30 DATA "ECUACIONES DE LA RECT
A","PARALELISMO Y
    PERPENDICULARIDAD","CAMBIOS
DE BASE","DISTANCIAS E
          INTERSECCIONES"
  40 GO SUB 500: IF CODE INKEY$<
49 OR CODE INKEY$>52 THEN GO TO
40
  50 GO TO 3000+100*VAL INKEY$
  70 DIM v(2,2): DIM m(9,2,2): D
IM u(2,2,2)
  85 GO TO 20
 175 IF NOT i(2,1)*i(1,2) THEN G
O TO 205
 246 IF NOT i(1,1)*i(1,2)*i(2,1)
*i(2,2) THEN GO TO 2700

 248 IF NOT e(2,1) THEN GO TO 28
0
 395 GO TO 2700
 400 LET a(1)=a(1)*SGN a(2): LET
 signo=SGN a(2): LET a(2)=ABS a(2
)
 700 IF NOT s(2,2) THEN GO TO 73
0
 710 LET p(1,1,1)=1: LET p(1,1,2
)=1: LET p(1,2,1)=0: LET p(1,2,2
)=1: LET p(2,2,1)=e(2,1): LET p(
2,2,2)=e(2,2): LET p(2,1,1)=0: L
ET p(2,1,2)=1
 720 IF s(1,2) THEN LET p(2,1,1)
=e(1,1): LET p(2,1,2)=e(1,2): LE
T p(2,2,1)=0: LET p(2,2,2)=1
 725 RETURN
 730 LET p(1,1,1)=0: LET p(1,1,2
)=1: LET p(1,2,1)=-e(2,1): LET p
(1,2,2)=e(1,1): LET p(2,1,1)=1:
LET p(2,1,2)=1: LET p(2,2,1)=0:
LET p(2,2,2)=1: RETURN

1425 IF NOT s(1,1)*s(2,1)*s(1,2)
*s(2,2) THEN GO SUB 700
1505 FOR i=1 TO 2: LET a(1)=i(i,
1): LET a(2)=i(i,2): GO SUB 400:
 LET i(i,1)=a(1): LET i(i,2)=a(2
): NEXT i
1605 PRINT AT 14,6;"5)Llamar mem
oria"
1630 GO SUB 500: IF CODE INKEY$<
49 OR CODE INKEY$>53 THEN GO TO
1630
1635 IF INKEY$="5" THEN GO TO 23
00
2300 INPUT "Introduce el numero
de la""memoria (1 a 9) ";memo
2310 IF memo<1 OR memo>9 OR memo
<>INT memo THEN GO TO 2300
2315 IF NOT (m(memo,1,2) OR m(me
mo,2,2) OR m(memo,1,1) OR m(memo
,2,1)) THEN PRINT #0;"Memoria va
cia": BEEP 1,-5: GO TO 1600
2320 GO SUB 2340: GO TO 100

2340 FOR i=1 TO 2: FOR j=1 TO 2:
 LET e(i,j)=m(memo,i,j): NEXT j:
 NEXT i: LET flag=2: GO SUB 1200
2350 RETURN
2700 PRINT #0;"Pulsa numero de m
emoria o volver": PAUSE 0
2710 IF CODE INKEY$>48 AND CODE
INKEY$<58 THEN LET memo=VAL INKE
Y$: GO TO 2730
2720 GO TO 20
2730 FOR i=1 TO 2: FOR j=1 TO 2:
 LET m(memo,i,j)=e(i,j): NEXT j:
 NEXT i: INPUT "": PRINT #0; FLA
SH 1;memo;" MEMORIZANDO": PAUSE
100: INPUT ""
2740 GO TO 2700
3100 GO TO 1600
3200 CLS : PRINT AT 4,2;"Opcione
s"''"1)Comprobar dos rectas"''
"2)Obtener una paralela"''"3)Obt
ener una perpendicular"''
3210 GO SUB 500: IF CODE INKEY$>
48 AND CODE INKEY$<52 THEN LET o
pc=VAL INKEY$: GO TO 3190+30*opc
3215 GO TO 3210
3220 GO SUB 3600
3230 IF m(re1,1,1)*m(re2,1,1)+m(
re1,1,2)*m(re2,1,2)=0 THEN PRINT
 "RECTAS PERPENDICULARES": GO SU
B 500: GO TO 3245
3240 PRINT "NI PARALELAS NI PERP
ENDICULARES": GO SUB 500
3245 GO TO 20
3250 INPUT "Memoria de la recta
a la que es paralela ";mia: LET
e(1,1)=m(mia,1,1): LET e(1,2)=m(
mia,1,2): GO TO 3700
3280 INPUT "Memoria de la recta
a la que es perpendicular ";mia:
 LET e(1,1)=-m(mia,1,2): LET e(1
,2)=m(mia,1,1): GO TO 3700
3300 GO TO 4000
3400 CLS : PRINT AT 4,2;"Opcione
s"''"1)Distancia punto-recta"''
"2)Distancia entre dos puntos"''
"3)Punto de corte de dos rectas
"''
3410 GO SUB 500: IF CODE INKEY$>
48 AND CODE INKEY$<52 THEN LET o
pc=VAL INKEY$: GO TO 3390+30*opc
3415 GO TO 3410
3420 INPUT "Memoria de la recta
 ";memo: GO SUB 2100: DIM z(2)
3425 LET z(1)=a(1): LET z(2)=a(2
): LET d$=e$: GO SUB 600: DIM y(
2): LET y(1)=a(1): LET y(2)=a(2)
3430 GO SUB 2340: LET dist=ABS (
i(1,1)*z(1)/i(1,2)/z(2)+i(2,1)*y
(1)/i(2,2)/y(2)+i(3,1)/i(3,2))
3435 LET dist=dist/SQR (i(1,1)*i
(1,1)/i(1,2)/i(1,2)+i(2,1)*i(2,1
)/i(2,2)/i(2,2))
3440 PRINT AT 17,0;"La distancia
 es ";dist: GO SUB 500: GO TO 20
3450 DIM z(2,2,2): FOR i=1 TO 2:
 GO SUB 2100: LET z(1,i,1)=a(1):
 LET z(1,i,2)=a(2)
3455 LET d$=e$: GO SUB 600: LET
z(2,i,1)=a(1): LET z(2,i,2)=a(2)
: NEXT i
3460 LET dist=ABS (z(1,1,1)/z(1,
1,2)-z(1,2,1)/z(1,2,2))↑2
3465 LET dist=dist+ABS (z(2,1,1)
/z(2,1,2)-z(2,2,1)/z(2,2,2))↑2
3470 LET dist=SQR dist: PRINT AT
 17,0;"La distancia es ";dist: G
O SUB 500: GO TO 20
3480 GO SUB 3600
3490 LET a(1)=m(re1,1,2)*m(re2,1
,2)*(m(re2,2,1)*m(re1,2,2)-m(re1
,2,1)*m(re2,2,2))
3500 LET a(2)=m(re1,2,2)*m(re2,2
,2)*(m(re1,1,1)*m(re2,1,2)-m(re2
,1,1)*m(re1,1,2))
3510 PRINT '"Coordenada x: ";:
GO SUB 7000
3515 IF ABS a(2)<>1 THEN LET a(1
)=a(1)*signo
3520 LET a(1)=m(re1,1,1)*a(1)*m(
re1,2,2)+m(re1,2,1)*m(re1,1,2)*a
(2): LET a(2)=a(2)*m(re1,1,2)*m(
re1,2,2)
3530 PRINT '"Coordenada y: ";:
GO SUB 7000: GO SUB 500: GO TO 2
0
3600 INPUT "Introduce las memori
as de ambas","recta 1: ";re1,"re
cta 2: ";re2
3610 IF m(re1,1,1)*m(re2,1,2)=m(
re2,1,1)*m(re1,1,2) THEN PRINT "
RECTAS PARALELAS": GO SUB 500: G
O TO 3245
3620 RETURN
3700 GO SUB 2100: GO SUB 1910: G
O SUB 1000+100*flag: GO TO 100
4000 CLS : PRINT AT 3,8;"Opcione
s"
4010 PRINT AT 6,0;"1)Introducir
nueva base"''"2)Cambiar un vecto
r de base"''''"   (ENTER vuelve)"
4020 GO SUB 500: GO TO 4020+80*(
CODE INKEY$=49)+180*(CODE INKEY$
=50)-4000*(CODE INKEY$=13)
4080 DATA "Componente x del prim
er vector","Componente y del pri
mer vector","Componente x del se
gundo vector","Componente y del
segundo vector"
4100 RESTORE 4080
4110 FOR i=1 TO 2: FOR j=1 TO 2
4120 READ a$: INPUT (a$), LINE d
$
4130 GO SUB 600: LET u(i,j,1)=a(
1): LET u(i,j,2)=a(2)
4150 NEXT j: NEXT i: GO TO 4000
4200 GO SUB 2200: LET v(1,1)=a(1
): LET v(1,2)=a(2): LET d$=e$: G
O SUB 600: LET v(2,1)=a(1): LET
v(2,2)=a(2)
4210 FOR i=1 TO 2: LET j=3-i: PR
INT AT 14+2*i,0;"Coordenada ";CH
R$ (119+i);": ";
4230 LET a(1)=(v(i,1)*v(j,2)*u(j
,j,1)*u(j,i,2)-v(j,1)*v(i,2)*u(j
,i,1)*u(j,j,2))*u(i,j,2)*u(i,i,2
)
4240 LET a(2)=(u(i,i,1)*u(j,j,1)
*u(j,i,2)*u(i,j,2)-u(j,i,1)*u(i,
j,1)*u(i,i,2)*u(j,j,2))*v(i,2)*v
(j,2)
4250 GO SUB 400: GO SUB frac
4260 NEXT i: GO SUB 500: GO TO 2
0
7000 GO SUB 400: IF ABS a(2)=1 T
HEN PRINT a(1)*SGN a(2);: LET le
n=LEN STR$ (a(1)*SGN a(2)): GO T
O 7130
7003 LET signo=SGN (a(1)*a(2))
```

CHR$ 18: Scroll izquierda de atributos. Se produce un scroll del archivo de atributos, una columna a la izquierda. El scroll puede ser lineal o esférico dependiendo de como se hubiera fijado previamente. Inicialmente es lineal. En el caso de scroll lineal, lo que va entrando por el lado contrario son los atributos permanentes en curso.

CHR$ 19: Scroll derecha de atributos. Se produce un scroll del archivo de atributos, una columna a la derecha. El scroll se produce de la misma forma que en el caso anterior.

CHR$ 20: Scroll abajo de atributos. Se produce un scroll del archivo de atributos, una línea hacia abajo. El scroll se produce de la misma forma que en los dos casos anteriores.

CHR$ 21: Scroll arriba de atributos. Se produce un scroll del archivo de atributos, una línea hacia arriba. El scroll se produce de la misma forma que en los tres casos anteriores.

CHR$ 22: Control «AT». Tiene dos operandos que son los dos caracteres que le siguen. El primero indica la línea y el segundo la columna, donde deberá quedar posicionado el cursor. El siguiente carácter se imprimirá en esa posición. Trabaja de la misma forma que el «AT» del Basic, salvo que el primer operando no podrá ser mayor de 21, aun cuando se hubieran eliminado las dos líneas inferiores de la pantalla.

CHR$ 23: Fija letra cursiva. Pone a «1» el flag de cursiva en la variable «FLAGS» del programa (no en la del Sistema Operativo con el mismo nombre).

CHR$ 24: Fija letra negrita (bold). Pone a «1» el flag de bold en la variable «FLAGS» del programa.

CHR$ 25: Fija letra especular (imagen de espejo). Pone a «1» el flag de espejo en la variable «FLAGS» del programa.

CHR$ 26: Fija letra normal. Pone a «0» los flags de cursiva, bold y espejo en la variable «FLAGS» del programa.

CHR$ 27: Fija scroll esférico. Pone a «1» el flag de scroll esférico en la variable «FLAGS» del programa.

CHR$ 28: Fija scroll lineal. Pone a «0» el flag de scroll esférico en la variable «FLAGS» del programa.

CHR$ 29: Transfiere la pantalla. Copia los archivos de pantalla y atributos en un bloque de 6912 bytes consecutivos, cuya primera dirección sea el contenido de la variable del Sistema «SEED». Al encender el ordenador, esta variable contiene «0», y cambia su valor cada vez que se hace uso de la función «RND» del Basic. Para almacenar el valor «n» en esta variable, basta con hacer: RANDOMIZE n donde «n» es un número entre «1» y «65535».

CHR$ 30: Recupera la pantalla. Copia, en los archivos de pantalla y atributos, el contenido de un bloque de 6912 bytes consecutivos, cuya primera dirección sea el contenido de la variable del Sistema «SEED».

CHR$ 31: Código nulo. No produce ningún efecto.

CHR$ 32 al CHR$ 255: Imprime el carácter que se corresponda con el código. La dirección del font de caracteres se toma de la variable del Sistema «CHARS». Es posible disponer un font de hasta 224 caracteres (el primero de los cuales deberá ser un espacio) en cualquier lugar de la memoria. Para direccionarlo, la variable del Sistema «CHARS» deberá contener un número que sea la dirección inicial del font, menos 256. La impresión se produce en letra normal, cursiva, negrita o especular (imagen de espejo), dependiendo de cómo estén fijados los flags correspondientes de la variable «FLAGS» del programa (no de la del Sistema que tiene el mismo nombre). Es posible imprimir, simultáneamente, en cursiva y negrita; pero si está a «1» el flag de «espejo», no se tendrán en cuenta los contenidos de los flags de cursiva y bold. Tras la impresión, se actualiza, adecuadamente, la posición de impresión. Asimismo, se actualizan los contenidos de las variables del Sistema «S-POSN» y «DF-CC» para que la rutina sea compatible con el Basic; en caso necesario, se produce un scroll automático de una línea. La impresión se realiza de modo «transparente», es decir, no se modifica el byte correspondiente del archivo de atributos. No se reconocen, como tales, los códigos de gráficos y tokens.

El programa «PROPAN» utiliza dos variables internas. La primera de ellas se denomina «FLAGS», tiene un octeto y contiene los 8 flags que necesita el programa. La segunda se llama «VAR-1», tiene 2 octetos y se utiliza para almace-

La siguiente instrucción a ejecutar será la que se encuentre en la dirección 7546h



OBJETO:

Si la condición «cc» es verdadera, carga en el registro contador de programa «PC» el último dato de la pila de máquina, poniendo el contenido de la dirección señalada por (SP) en el octeto inferior y la dirección señalada por (SP)+1 en el octeto superior, después incrementa en 2 el registro «SP». Con esto efectúa la vuelta desde una subrutina a la instrucción siguiente a una «CALL» que realizó la entrada. Si la condición «cc» es falsa, ejecuta la siguiente instrucción. Los códigos nemotécnicos de «cc» y sus valores binarios se indican a continuación.

«cc»	binario	significado
NZ	000	no cero
Z	001	cero
NC	010	no acarreo
C	011	acarreo
PO	100	paridad impar o no desbordamiento
PE	101	paridad par o desbordamiento
P	110	positivo
M	111	negativo

CODIGO DE MAQUINA:

1 1 ← cc → 0 0 0

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

Si la condición «cc» es verdadera:

3

Si la condición «cc» es falsa:

1

CICLOS DE RELOJ:

Si la condición «cc» es verdadera:

11

Si la condición «cc» es falsa:

5

EJEMPLO:

RET Z

Indicador de condición Z=1

Contenido del registro «SP»

(SP):	1 1 1 1 0 1 1 1	F7h
	0 0 0 0 0 1 0 0	04h

Contenido de los octetos F704h y F705h

F704h:	0 0 0 0 0 0 1 1	03h
F705h:	0 1 1 1 0 0 0 0	70h

Instrucción

RET Z:	1 1 0 0 1 0 0 0	C8h

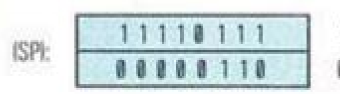
Contenido del registro «SP» después de la ejecución

(SP):	1 1 1 1 0 1 1 1	F7h
	0 0 0 0 0 1 1 0	06h

Contenido del registro «PC» después de la ejecución



La siguiente instrucción a ejecutar será la que se encuentra en la dirección 7003h



OBJETO:

Retorna desde una interrupción enmascarable al punto por donde iba el programa cuando se recibió la petición de interrupción («INT»). Lo realiza cargando en el registro contador de programa «PC» el último dato de la pila de máquina, poniendo el contenido de la dirección señalada por (SP) en el octeto inferior y la dirección señalada por (SP)+1 en el octeto superior. A continuación incrementa en 2 el registro «SP».

Todos los chips de periféricos Z-80 reconocerán la ejecución de esta instrucción para su propio control y poder encajar las interrupciones; para ello, el microprocesador envía una señal de acuse de interrupción poniendo a «0», de forma simultánea, las patas «IORQ» y «M1».

Esta instrucción no actúa sobre los flip-flops IFF1 e IFF2 que son los que controlan la prioridad de las interrupciones.

CODIGO DE MAQUINA:

1 1 1 0 1 1 0 1	EDh
0 1 0 0 1 1 0 1	4Dh

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

4

CICLOS DE RELOJ:

14

EJEMPLO:

RETI

Contenido del registro «SP»

(SP):	11110111	F7h
	01000110	46h

Contenido de los octetos F746h y F747h

F746h:	11010100	D4h
F747h:	01101001	69h

Instrucción

RETI:	11101101	EDh
	01001101	4Dh

Contenido del registro «SP» después de la ejecución

(SP):	11110111	F7h
	01001000	48h

Contenido del registro «PC» después de la ejecución

(PC):	01101001	69h
	11010100	D4h

La siguiente instrucción a ejecutar será la que se encuentra en la dirección 69D4h, es decir, la siguiente a aquella que se estaba ejecutando cuando se recibió la petición de interrupción.

RETN

OBJETO:

Retorna desde una interrupción no enmascarable al punto por donde iba el programa cuando se produjo la petición de interrupción. Para ello, carga en el registro contador de programa «PC» el último dato de la pila de máquina, poniendo el contenido de la dirección señalada por (SP) en el octeto inferior y la dirección señalada por (SP) + 1 en el octeto superior, después incrementa en 2 el registro «SP».

Además copia el estado del flip-flop IFF2 en el IFF1, con lo que éste recupera el valor que tenía antes de producirse la interrupción no enmascarable.

CODIGO DE MAQUINA:

11101101	EDh
01000101	45h

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

4

CICLOS DE RELOJ:

14

EJEMPLO:

RETN

Contenido del registro «SP»

(SP):	11110111	F7h
	01001000	48h

Contenido de los octetos F748h y F749h

F748h:	00101101	2Dh
F749h:	01010110	56h

IFF1 = Ø, IFF2 = 1

Instrucción

RETN:	11101101	EDh
	01000101	45h

Contenido del registro «SP» después de la ejecución

(SP):	11110111	F7h
	01001010	4Ah

IFF1 = 1 y IFF2 = 1, después de la ejecución

Contenido del registro «PC» después de la ejecución

(PC):	01010110	56h
	00101101	2Dh

La siguiente instrucción a ejecutar será aquella que se encuentre en la dirección 562Dh, es decir, la siguiente a la que se estaba ejecutando cuando se recibió la petición de interrupción.

Observaciones:

Para entender y manejar mejor estas instrucciones es interesante que tenga en cuenta lo siguiente:

Las interrupciones son formas de trabajo que se utilizan principalmente para atender requerimientos que no admiten demora, resulta fundamental poder volver a la secuencia del programa que se estaba ejecutando una vez atendido ese requerimiento. Lo que garantizan las instrucciones RETI y RETN es la vuelta a la instrucción donde se interrumpió el programa.

Las instrucciones RETI o RETN serán siempre la última

Por otro lado, los códigos de control van a permitirnos borrar la pantalla por trozos, intercambiar zonas, repetir «n» veces un carácter, mover el cursor, realizar «scroll» de pantalla o atributos tanto en modo esférico como en modo lineal e, incluso, transferir la pantalla completa a otra dirección de memoria o recuperarla desde allí. Asimismo, se reconocen los códigos de «AT» y «ENTER» que funcionan de la misma manera que en Basic.

A continuación, se da una lista detallada de todos los códigos que utiliza «PROPAN»:

CHR$ Ø: Código nulo, no produce ningún efecto.

CHR$ 1: Borra el primer tercio de pantalla. El borrado se produce poniendo a «Ø» los bits del archivo de pantalla y copiando los atributos permanentes en los bytes del archivo de atributos. No se restaura la posición de impresión.

CHR$ 2: Borra el segundo tercio de pantalla. El borrado se produce de la misma forma que en el caso anterior.

CHR$ 3: Borra el tercer tercio de pantalla. El borrado se produce de la misma forma que en los dos casos anteriores.

CHR$ 4: Intercambia el primer tercio de la pantalla con el segundo. Se intercambian tanto los bytes del archivo de pantalla como los del de atributos. La posición de impresión no resulta afectada.

CHR$ 5: Intercambia el segundo tercio de pantalla con el tercero. El intercambio se produce de la misma forma que en el caso anterior.

CHR$ 6: Intercambia el primer tercio de pantalla con el tercero. El intercambio se produce de la misma forma que en los dos casos anteriores.

CHR$ 7: Repetición de carácter. Utiliza dos operandos que serán los dos caracteres que le siguen; el primero de ellos indica las veces que se ha de repetir el carácter y el segundo es el carácter a repetir. Por ejemplo:

```
CHR$ 7;CHR$ 15;«*»;
```

provocaría la impresión de 15 asteriscos a partir de la posición actual de impresión. El segundo operando nunca puede ser «7», si lo fuera, la orden sería ignorada. Por ejemplo:

```
CHR$ 7;CHR$ 15;CHR$ 7;
```

no provocaría ningún efecto. La posición de impresión se actualiza según el número de caracteres que se impriman. La rutina trabaja en modo «recursivo».

CHR$ 8: Cursor izquierda. Al igual que en Basic, retrocede una columna la posición de impresión. A diferencia del Basic, no va más allá de la columna «Ø».

CHR$ 9: Cursor derecha. Avanza una columna la posición de impresión. No va más allá de la columna «31».

CHR$ 1Ø: Cursor abajo. Baja una línea la posición de impresión. No va más allá de la línea «21».

CHR$ 11: Cursor arriba. Sube una línea de la posición de impresión. No va más allá de la línea «Ø».

CHR$ 12: DELETE. Produce el borrado del carácter anterior a la actual posición de impresión. El borrado se produce retrocediendo el cursor, imprimiendo un espacio y volviendo a retroceder el cursor. Si la posición de impresión se encontrara en «Ø,Ø», se borraría el carácter situado en «Ø,Ø» quedando la posición de impresión, de nuevo, en «Ø,Ø». En cualquier otra posición de impresión, se borra el carácter anterior y la posición de impresión no varía.

CHR$ 13: ENTER. Se avanza la posición de impresión al inicio de la siguiente línea. Si se estuviera en la última línea, se produce un «scroll» automático de una línea. Hay que tener en cuenta que el Basic manda, automáticamente, este código al final de cualquier comando «PRINT» o «LPRINT» que no termine en «;».

CHR$ 14: Scroll izquierda de pantalla. Se produce un «scroll» de la pantalla, un pixel a la izquierda. El scroll puede ser «lineal» o «esférico» dependiendo de como se hubiera fijado previamente. Al cargar el programa, queda preparado para «scroll lineal».

CHR$ 15: Scroll derecha de pantalla. Se produce un scroll de la pantalla, un pixel a la derecha. El scroll se produce de la misma forma que en el caso anterior.

CHR$ 16: Scroll abajo de pantalla. Se produce un scroll de la pantalla, un scan hacia abajo. El scroll se produce de la misma forma que en los dos casos anteriores.

CHR$ 17: Scroll arriba de pantalla. Se produce un scroll de la pantalla, un scan hacia arriba. El scroll se produce de la misma forma que en los tres casos anteriores.

ma en código máquina requiere un planteamiento bastante más cuidadoso que si se hiciera en Basic. Es importante definir, con toda claridad, qué es lo que se quiere hacer y cómo va a hacerse. Por pequeño que sea el programa, es necesario escribir un gran número de líneas y las posibilidades de error aumentan enormemente. Por todo esto, es recomendable que todo programa en código máquina esté compuesto por una colección de rutinas que serán llamadas desde el bucle principal para realizar cada una de las funciones elementales. Estas rutinas pueden escribirse y probarse, previamente, por separado. De esta forma, será más sencillo detectar y corregir los posibles errores.

A lo largo del curso, hemos ido viendo un gran número de rutinas sueltas. Algunas de ellas, trabajaban sobre la pantalla y, entre éstas, algunas realizaban una función u otra dependiendo del número que contuviera el acumulador al entrar en ellas (borrado parcial, intercambio de zonas de pantalla, etc.). También vimos lo que era un canal de salida; una rutina que hacía algo con el número que contuviera el acumulador al entrar en ella (por ejemplo, imprimir el carácter cuyo código fuera ese número).

Ahora, vamos a agrupar todas estas rutinas de gestión de pantalla, y unas cuantas más, para crear un auténtico «Procesador de Pantalla» que trabaje como un canal de salida. Es decir, en nuestro programa se entrará con un número en el acumulador y, dependiendo de este número, el programa realizará una u otra función sobre la pantalla. Si almacenamos la dirección de inicio de nuestro programa en la tabla de canales, podremos utilizarlo desde el Basic, como si se tratara de uno de los canales del Sistema Operativo. La utilidad de este programa es que va a permitirnos un gran número de efectos sobre la pantalla, tanto si programamos en Basic como si lo hacemos en código máquina.

El programa —al que, por razones evidentes, hemos denominado «PROPAN»— utiliza 3Ø códigos de control (del «1» al «3Ø») que realizan distintas funciones. Los códigos «Ø» y «31» son ignorados y no realizan ninguna función, pero cabe la posibilidad de que el lector amplíe el programa utilizando estos códigos para realizar otras funciones que él incorpore. Finalmente, los códigos del «32» en adelante (hasta el «255») producen la impresión de caracteres. Si se utiliza el «font» de caracteres de la ROM, sólo se podrán utilizar los caracteres del «32» al «127», pero nada impide crear otro «font» de 223 caracteres con códigos comprendidos entre «32» y «255» (el «32» tiene que ser, siempre, un espacio) y direccionario mediante la variable del Sistema «CHARS» (dirección 236Ø6 y 236Ø7); con lo que no estaremos restringidos a los 21 UGDs del Spectrum.

Los caracteres se podrán imprimir tanto en letra normal como en cursiva o negrita (bold) y también es posible imprimirlos en «imagen de espejo»; esto último es muy útil, ya que nos ahorra mucho trabajo en la definición de gráficos; si tenemos un carácter que representa a un muñeco corriendo hacia la derecha, bastará imprimirlo en «imagen de espejo», para que aparezca corriendo hacia la izquierda.

REINICIOS DE PAGINA CERO

Código Fuente	Hexadecimal	Decimal
RST #ØØ	C7	199
RST #Ø8	CF	2Ø7
RST #1Ø	D7	215
RST #18	DF	223
RST #2Ø	E7	231
RST #28	EF	239
RST #3Ø	F7	247
RST #38	FF	255

Fig. 11-6. Tabla de codificación para los reinicios de página cero.

instrucción de las rutinas que atiendan las interrupciones o de la rutina que trata la interrupción no enmascarable respectivamente, si no fuera así, no se podría trabajar con esta técnica.

Los flip-flops IFF1 e IFF2 son, desde el punto de vista software, dos flags que controlan cuando se permiten o no las interrupciones. IFF1 cuando vale 1 las permite y cuando vale cero no, IFF2 es el almacenamiento temporal del valor de IFF1.

Las interrupciones no enmascarables no están afectadas por el valor de IFF1 o IFF2, siempre se permiten.

En el programa monitor del SPECTRUM está anulada la interrupción no enmascarable. Si se recibiera una petición de este tipo, la rutina de servicio se limita a retornar a la secuencia principal. El objeto de esta medida es facilitar la protección de software comercial, ya que un programador experto, podría entrar en un programa protegido, provocando, desde fuera, una interrupción no enmascarable y saltando a una rutina que le devolviera el control.

Por otro lado, la ULA genera, cada 2Ø milisegundos, una petición de interrupción enmascarable que será atendida o no en función del estado del flip-flop IFF1. En el capítulo que trata de las instrucciones de control de la CPU, veremos detalladamente cómo funcionan las interrupciones y cómo utilizarlas en nuestros programas.

Reinicios de página cero

En todo programa, y especialmente en un Sistema Operativo, existen una serie de rutinas a las que hay que llamar frecuentemente desde distintos puntos del programa —imprimir un carácter, realizar un cálculo, leer un carácter de memoria, etc.—. Esto obligaría a utilizar un gran número de instrucciones «CALL» dirigidas al mismo punto desde varios lugares del programa. Cada instrucción «CALL» ocupa tres bytes, por tanto, el gasto de memoria resulta considerable.

Para evitar este desperdicio de memoria, el Z-8Ø incorpora una instrucción que permite llamar a subrutinas colocadas en lugares fijos de la memoria. Cada una de estas llamadas sólo ocupa un byte, de forma que se reduce considerablemente el gasto de memoria. Esta instrucción tiene el nemónico «RST», abreviatura del Inglés «ReStart» (Reinicio).

Las rutinas llamadas por esta instrucción han de estar situadas en las primeras direcciones de memoria, por ello, esta instrucción se denomina «Reinicio de página cero».

Todos los «Restart» del Spectrum se dirigen a rutinas útiles de la ROM que podremos utilizar en nuestros programas. De hecho, ya hemos utilizado una de ellas. ¿Recuerda el lector la misteriosa instrucción «RST #Ø8» que nos detenía el programa con un informe de error? Pues bien, existen un total de 8 reinicios de página cero y cada uno de ellos tiene su utilidad. En principio, vamos a ver la instrucción y, luego, estudiaremos cada una de las rutinas a las que podemos llamar.

RST p

OBJETO:

Lo primero que hace el micro-procesador, al igual que en todas las instrucciones, es incrementar el registro contador de programa «PC» en el número de octetos que tiene la instrucción, en este caso uno. Después guarda el valor del registro «PC» en la pila de máquina poniendo el contenido del octeto superior en la dirección señalada por (SP)-2. A continuación decrementa en 2 el registro «SP». Finalmente carga en el registro «PC» la dirección de página cero indicada por el operando «p». La correspondencia para codificar la instrucción es como sigue:

«p»	«t»
ØØh	ØØØ
Ø8h	ØØ1
1Øh	Ø1Ø
18h	Ø11
2Øh	1ØØ
28h	1Ø1
3Øh	11Ø
38h	111

con lo que comenzará a ejecutar las instrucciones que se encuentren en dichas direcciones.

CODIGO DE MAQUINA:

1 1 ← t → 1 1 1

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

3

CICLOS DE RELOJ:

11

EJEMPLO:

```
RST #18
```

Dirección del primer octeto de la instrucción

01010010	52h
10000011	83h

Contenido del registro «SP»

(SP):	11110000	F0h
	00111100	3Ch

Instrucción

RST #18:	11011111	DFh

Contenido del registro «SP» después de la ejecución

(SP):	11110000	F0h
	00111010	3Ah

Contenido de los octetos F03Ah y F03B después de la ejecución

F03Ah:	10000100	84h
F03Bh:	01010010	52h

Contenido del registro «PC» después de la ejecución

(PC):	00000000	00h
	00011000	18h

La siguiente instrucción a ejecutar está en la dirección 0018h

El programa monitor del SPECTRUM tiene las siguientes rutinas de página cero programadas:

Dirección 0000h

START:

Inicializa toda la memoria. Produce el mismo efecto que apagar y encender la fuente de alimentación o pulsar el botón de RESET en el SPECTRUM PLUS. Sería el equivalente, en código máquina, al famoso «RANDOMIZE USR 0» del Basic que nos borra toda la memoria.

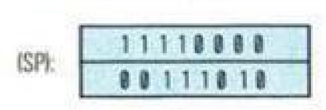
Dirección 0008h

ERROR-1:

Es la rutina de manejo de errores. Cuando se utilice, el Sistema Operativo detendrá la ejecución de cualquier programa y saltará al editor de Basic presentando, en la parte inferior de la pantalla, un mensaje de error del Sistema. El mensaje presentado dependerá del contenido del byte siguiente a esta instrucción. Este byte se denomina «literal». En la siguiente tabla se da una lista de todos los mensajes del Sistema en función del literal que siga a «RST #08».

literal	mensaje	
255	0	OK
0	1	NEXT without FOR
1	2	Variable no fount
2	3	Subscript wrong
3	4	Out of memory
4	5	Out of screen
5	6	Number too big
6	7	RETURN without GOSUB
7	8	End of file
8	9	STOP statement
9	A	Invalid argument
10	B	Integer out of range
11	C	Nonsense in BASIC
12	D	BREAK - CONT repeats
13	E	Out of DATA
14	F	Invalid file name
15	G	No room for line
16	H	STOP in INPUT
17	I	FOR without NEXT
18	J	Invalid I/O device
19	K	Invalid colour
20	L	BREAK into program
21	M	RAMTOP no good
22	N	Statement lost
23	O	Invalid stream
24	P	FN without DEF
25	Q	Parameter error
26	R	Tape loading error
27	"	
28		(c) 1982 Sinclair Research Ltd

Por ejemplo, la secuencia para detener un programa de forma que aparezca, en pantalla, el mensaje «R Tape loading error», sería:



```
RST #8
DEFB 26
```

Cuando expliquemos el uso del Microdrive en código máquina, veremos que es posible utilizar esta instrucción, con otros literales, para acceder a las funciones del INTERFACE 1. De momento, en la versión básica del Spectrum, si utilizamos otros literales, obtendremos informes sin sentido.

Dirección 0010h

PRINT-A-1

Imprime el carácter o código de control cuyo valor en ASCII se pondrá previamente en el registro acumulador «A». Antes de ello, es necesario abrir un canal de comunicación, lo cual se consigue llamando a la rutina «CHAN-OPEN» de la ROM (dirección: 1601h) con el número de corriente en el acumulador. En resumen, se puede decir que ésta es la rutina general de acceso a todos los canales de salida.

Dirección 0018h

GET-CHAR

Comprueba si el carácter de la posición de memoria direcciónada por la variable CH-ADD es imprimible, en caso afirmativo retorna; en otro caso incrementa la variable CH-ADD hasta que encuentre un carácter imprimible o el código «ENTER».

Dirección 0020h

NEXT-CHAR

Realiza la misma operación que la rutina anterior (GET-CHAR) comenzando por el carácter siguiente al direccionado por la variable CH-ADD.

Dirección 0028h

FP-CALC

Esta es la entrada de la rutina del calculador. Dada la complejidad de esta rutina, le dedicaremos un capítulo aparte. No obstante, anticiparemos que las operaciones a realizar se indican, también, mediante literales. Cuando aprendamos a manejar esta rutina, tendremos a nuestra disposición toda la potencia de cálculo del Basic para usarla desde código máquina.

Dirección 0030h

BC-SPACES

Deja un número de posiciones libres en el área de trabajo desplazando el resto de la memoria hacia arriba. El número de posiciones viene dado por el contenido del par de registros «BC» al entrar en la rutina.

Dirección 0038h

MASK-INT

A esta posición salta el micro procesador cuando se produce una interrupción enmascarable (en MODO 1), lo cual ocurre en el SPECTRUM cada 20 milisegundos.

Esta rutina se utiliza para actualizar el reloj de tiempo real y para leer el teclado. El código ASCII del carácter, dígito o signo pulsado lo devuelve esta rutina en la variable «LAST-K». También es posible llamarla desde nuestros programas para leer el teclado.

INSTRUCCIONES DE LLAMADA Y RETORNO

Código Fuente	Hexadecimal	Decimal
CALL nn	CD,n,n	205,n,n
CALL C,nn	DC,n,n	220,n,n
CALL NC,nn	D4,n,n	212,n,n
CALL Z,nn	CC,n,n	204,n,n
CALL NZ,nn	C4,n,n	196,n,n
CALL PE,nn	EC,n,n	236,n,n
CALL PO,nn	E4,n,n	228,n,n
CALL M,nn	FC,n,n	252,n,n
CALL P,nn	F4,n,n	244,n,n
RET	C9	201
RET C	D8	216
RET NC	D0	208
RET Z	C8	200
RET NZ	C0	192
RET PE	E8	232
RET PO	E0	224
RET M	F8	248
RET P	F0	240
RETI	ED,4D	237,77
RETN	ED,45	237,69

Fig. 11-5. Tabla de codificación para las instrucciones de llamada y retorno.

Tablas de codificación

En la Figura 11-5 está la tabla de codificación para las instrucciones de llamada y retorno, y en la 11-6, para los reinicios de página cero.

Ejemplos

Terminado este capítulo, sólo nos restan por estudiar las instrucciones de entrada/salida y las de control de CPU. No obstante, estamos ya en disposición de escribir un auténtico programa en código máquina.

La realización de un progra-

COMO SE CONSTRUYE UN PROGRAMA COMERCIAL

José Manuel MUÑOZ

Esta historia empieza en un mundo muy viejo, bañado por la luz de una pequeña estrella más vieja aún. En un tiempo y un espacio muy alejados de éste y en el que regían leyes físicas muy diferentes a las de hoy.

En aquel mundo había grandes ciudades amuralladas, cuyas altas torres de piedra roja se elevaban hasta el cielo, y salvajes dragones que infestaban las montañas. Había tierras inexploradas llenas de criaturas desconocidas y muchas razas distintas habitaban los dos vastos continentes del Planeta, Althassir y Vramtis, que se hallaban en guerra entre sí desde hacía tanto tiempo, que ninguna criatura viviente recordaba lo que era la paz.

Todo Althassir temblaba bajo el yugo mágico de Zwolhan, jefe supremo del Círculo de la Esfera Resplandeciente, y Vramtis hacía lo propio bajo Kulwoor, el señor del Castillo bajo la Montaña. Ambos brujos se hacían la guerra con la esperanza de llegar a dominar todo el planeta, pero tal cosa no parecía fácil ya que sus poderes estaban demasiado equilibrados. El temor a ambos hechiceros era tal, que nadie pensaba en rebelarse. Es cierto que había antiguas leyendas que hablaban de ejércitos enteros que se habían aventurado dentro del Castillo bajo la Montaña, pero eran bastante ambiguas sobre si habían vuelto o no a salir.

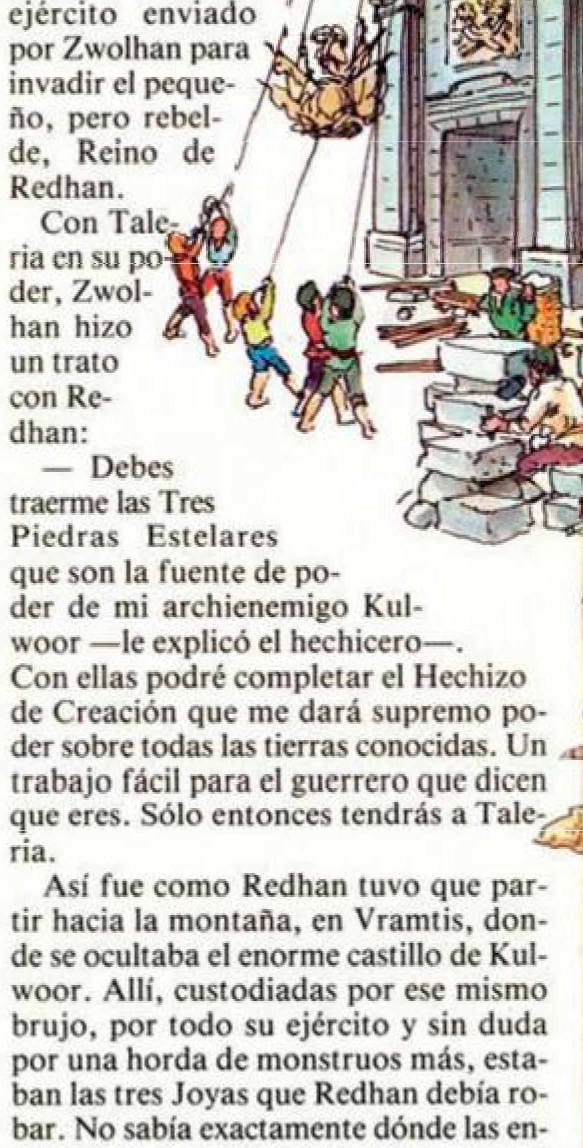
Había, sin embargo, un guerrero que no se sometía. Su nombre era Redhan y aunque ahora llevaba una sencilla cota de mallas, en otros tiempos sus hombros habían llevado la Túnica Blanca de los Señores. Sus únicas posesiones eran ahora una sencilla y vieja armadura, llena de golpes que hablaban de muchas batallas, y una negra y temible espada que Zwolhan le había entregado.

Cuando recordaba lo que había perdido, Redhan necesitaba de todo el dominio de sí mismo para que la ira no le dominase. Sin embargo, su sueño no consistía en recuperar su Reino, sino en rescatar a su amada Taleria de las Tierras Rojas, de las garras de Zwolhan. Recordaba cómo una horda de Morfocambiantes, disfradados como guerreros de las tierras de Taleria, habían entrado en su castillo y derrotado a la pequeña guarnición.

Entonces había logrado huir con Taleria y un grupo de guerreros, mientras el castillo ardía a sus espaldas. Pero en el camino habían sido capturados por un ejército enviado por Zwolhan para invadir el pequeño, pero rebelde, Reino de Redhan.

Con Taleria en su poder, Zwolhan hizo un trato con Redhan:

— Debes traerme las Tres Piedras Estelares que son la fuente de poder de mi archienemigo Kulwoor —le explicó el hechicero—. Con ellas podré completar el Hechizo de Creación que me dará supremo poder sobre todas las tierras conocidas. Un trabajo fácil para el guerrero que dicen que eres. Sólo entonces tendrás a Taleria.



Así fue como Redhan tuvo que partir hacia la montaña, en Vramtis, donde se ocultaba el enorme castillo de Kulwoor. Allí, custodiadas por ese mismo brujo, por todo su ejército y sin duda por una horda de monstruos más, estaban las tres Joyas que Redhan debía robar. No sabía exactamente dónde las encontraría, pero era casi seguro que al menos una de ellas estaría defendida por Glaurung, el Dragón aliado de Kulwoor y que habitaba en las profundidades de la montaña, bajo el castillo.

Y aquí precisamente comienza su aventura. Redhan, haciéndose pasar por un simple mercenario, logra entrar en el castillo. Pero salir de él por la puerta situada en el ala opuesta de la montaña, con las tres Joyas que hay que arrebatar a Glaurung y a Kulwoor, será mucho más difícil.

Si lo logra, o no, dependerá de nosotros.

Volviendo a la realidad

Tal vez una de las fases más sencillas y tranquilas en la creación de un juego,



es la elaboración de la idea en la que va a basarse. Esta idea puede venir de cualquier parte; de una historia, de una película, de un cómic o incluso —por qué no decirlo— de otro programa. Aquí, sin embargo, se corre el peligro de entusiasmarse excesivamente con el diseño y hacerlo crecer peligrosamente. Y así, cuando el padre de la criatura la presenta ante sus compañeros de trabajo, suele ocurrir que...

—Verás, chaval..., todo esto está muy bien, —le dicen ellos sonriendo maliciosamente— pero hay un pequeño problema.

—¿Cuál?, ¿cuál?...

—El Spectrum no tiene un Mega de memoria interna. —Le sueltan despiadadamente.

Entonces llega el temible momento de los tijeretazos. Todos los componentes del equipo comenzamos a depurar el esquema del juego hasta dejarlo reducido a límites aceptables para el Spectrum. Una vez hecho esto hay que decir qué rutinas se necesitan y lo que éstas deben hacer y cómo.

Y ya, por fin, terminada la fase de análisis, corren ríos de tinta, se generan kilómetros de listados y se preparan las primeras pruebas.

Al llegar a este momento, el programa aún está lejos de terminarse puesto que hay que eliminar muchos fallos de sintaxis y de lógica. Estos últimos son, con mucho, los más difíciles de encontrar. Por ejemplo, en la primera prueba que se hizo de nuestra videoaventura, vimos asombrados cómo el sprite de nuestro personaje se hundía en el suelo, desapareciendo de la pantalla. Al hacerlo, pasó a la pantalla inferior del mapa y siguió cayendo vertiginosamente durante un rato. Siguió así hasta que todo se llenó de colorines y apareció el conocido mensaje del Copyright de la ROM del Spectrum.

En resumen: el sistema había cascado. ¿Por qué? Habría que rastrear el error y eliminarlo antes de seguir adelante.

—Creo que el error va a estar en DETECC, —dijo uno de los compañeros—. No detecta el suelo y por eso el sprite se cae.

Alzamos las cejas. DETECC es una rutina utilizada en el programa para examinar el retorno de todos los personajes en movimiento y es ahora nuestra. Con ella se detectan los choques con los objetos sólidos..., y se producen las caídas cuando no hay nada bajo los pies de los personajes.

—DETECC está bien, —dijimos riéndonos— la culpa la va a tener tu rutina de LIMITES de cambio de pantalla. Es una ensalada inconexa de bytes con revuelto de bugx, una porquería entrópica, una montaña de...

LISTADO 1

```
  5 REM  DEMO "GIGANTULA"
ARTICULO MICROHOBBY
  7 REM  PARA DETENER LA
DEMO MANTENER PULSADA
CUALQUIER TECLA
 10 CLEAR 29999
 20 LOAD ""CODE 30000,991
 30 RANDOMIZE USR 30000
```

LISTADO 2

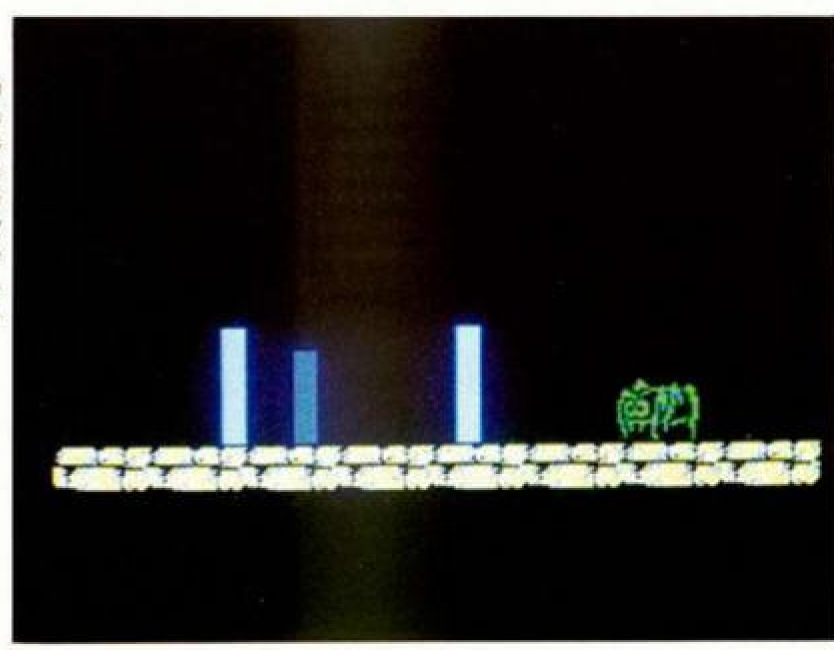
1	CD8E75CDF075CD1D76AF	1553
2	DBFEFEFF28F3FEBF28EF	1989
3	C9C9264078E618846778	1233
4	E6070F0F0F816FC9DD46	1014
5	02DD4E03DD7E01F5DD7E	1244
6	00F5CD4675C5CB79200E	1204
7	CB78200A06081A771324	579
8	10FA1805060813 10FDC1	790
9	F1043D20DEF1DD46020C	1106
10	3D20D2C9210040110140	683
11	36000100 18EDB0364401	615
12	FF02EDB03E00D3FEDD21	1451
13	3F76DD36030011477606	671
14	08D5C5CD5675C1D1DD7E	1575
15	03C604DD770310EF0640	873
16	21005A36702310FB2167	727
17	593E6D0605771 10A0019	442
18	77111600 1910F4218A59	703
19	3E4C0604771120001910	357
20	F9C9DD214376DD3602FC	1418
21	DD36031B0611118D760E	618
22	04D5C5CD567501E8030B	1069
23	78B120F8C1D11B1B0D20	1081
24	ECDD340210E2C9DD3602	1231
25	0C0620118776 0E04D5C5	748
26	CD567576 76C1D121A000	1239
27	19EB0D20EFDD350310E5	1066
28	C9020410200 4050CFBFF	782
29	E0D4A0C0A0C0FF1F2E1D	1501
30	280C08A8FFFF070A020A	767
31	1628FF50A0000000010A8	744
32	40FF83050321C101FF0F	955
33	0B072A12020203FF8005	473
34	022040A143FC0A150200	611
35	00413300000000000000	116
36	00000000000000000000	0
37	0103070C181818101010	143
38	08080000000000000000	16
39	00000000000000000021	33
40	73F9DE820D1D12121F2D	870
41	302F2D2D454000000000	318
42	00000000000000000000	0
43	C0E09F618307C6E46C64	1444
44	E4C008F7C18000000000	996
45	00000000000000000000	0
46	00000001F3089F8FBBC3	984
47	C7C7E6666C7C3C2C2828	1146
48	28000000000000000000	40
49	000000C0E0E070703818	944
50	18303020206040404000	472
51	00000000000000000000	0
52	00000000000000000000	0
53	00000101010101010101	8
54	00000000000000000000	0
55	00000000000000000101	2
56	43E2FD849BBBA4A4BF98	1694
57	A09F998989800000000 0	874
58	00000000000000000002	2
59	86C73FCF0F1B9BDADAD2	1446
60	D0B030E9A72020000000	896
61	00000000000000000000	0
62	0000021 7BF3B3D7D7646	649
63	0606060637F703020202	335
64	01000000000000000000	1
65	00000000000000008080	256
66	80808080808080000000	896
67	00000000000000000000	0
68	00000000000000000101	2
69	01030202020202020301	20
70	01010000000000000000	2
71	0000000000000 0C0E1F98	209
72	C3F4106EEF9393FF6E80	1591
73	7F6624A4800000000000	685
74	00000000000000000000	0
75	09FF191D3D3D2D2B2B23	606
76	6666EC65444444060200 0	869
77	00000000000000000000	0
78	00808CDECE4667A7A303	1202
79	03030317F68606020202	424
80	02000000000000000000	2
81	00000000000000000000	0
82	00000000000000000000	0
83	00000000000000000000	0
84	0000000000060E0F1F1B	93
85	1B30736744C487838403	958
86	03030100000000000000	7
87	0000000000000000181C	52
88	3798A08171799B99F970	1399
89	02FD7060400000000000	527
90	00000000000000000000	0
91	0000FD46E7EFEB6B6161	1329
92	616367DFC6C444444400	1120
93	00000000000000000000	0
94	00000040E0F030B0B898	1088
95	9898989808 0C0C040404	652

Se nos fue la voz al venirnos a la mente un terrible presentimiento. Agarramos el listado de la rutina y allí, en un registro equivocado, estaba el fallo. Lo arreglamos mientras nuestros compañeros silbaban con disimulo.

—El primero que diga algo, se traga el microdrive, —dijimos muy serios.

Después de esta pequeña disertación sobre las alegrías de la programación, nos centraremos más en los detalles de un programa en concreto; el nuestro.

Nuestro programa es una videoaventura en la que aparecen guerreros de distintos tipos, monstruos variados e incluso un curioso dragón volador. Está centrado en un castillo enterrado bajo la cima de una montaña y los escenarios,



La rutina en código máquina que os ofrecemos podréis observar la perfección de movimientos y gráficos de uno de los protagonistas. «Gigántula».



En el lugar de trabajo de los protagonistas observaremos una frenética actividad.

aunque bastante variopintos, son los propios de un castillo con subterráneos. Hay personajes que andan, otros que vuelan, que arrojan hechizos o bolas de fuego, etc... ¿Por qué pensamos que es diferente de otros programas que también tienen castillos, guerreros, magos y otras cosas por el estilo?

Las razones que nos llevaron a codificar este programa y no otro cualquiera, son bastantes e irán siendo reveladas a medida que nos introduzcamos un poco más en la idea del juego. De momento nos limitaremos a decir que un punto muy importante para nosotros era que las acciones necesarias para superar una pantalla o concluir el juego, nunca fuesen las mismas exactamente de una vez para otra. En muchos programas, para superar una pantalla, debemos aprender a saltar o echar a andar en un momento determinado, ya que nuestros enemigos no se percatan realmente de nuestra existencia y tienen un ciclo de movimientos repetitivo y, por tanto, previsible. En otros hay que aprender que primero hay que ir a la pantalla tal, para coger algo, que nos ayudará a hacer otra cosa que, a su vez, nos servirá para..., y así hasta el final. En ambos casos, una vez que conozcamos el esquema fijo para superar las distintas pantallas, el juego no tendrá historia.

Eso nos llevó a pensar en hacer un programa que nunca fuese exactamente igual cada vez que se jugase. Esto no quiere decir que cambiemos el mapa cada vez que se juegue. Estuvimos tentados de hacerlo, pero al final llegamos a la conclusión de que los «mapófilos» no nos lo perdonarían.

No, lo que cambiamos son ciertos detalles como la colocación de los personajes en las pantallas, número y tipo y la colocación en éstas de objetos que pueden ayudar a fastidiar, según el caso, con lo cual, la ruta a seguir no tiene por qué ser siempre la misma. Además, los guerreros enemigos, que pueden andar o saltar o luchar al igual que nuestro personaje, no se mueven al azar o siguiendo un ciclo, sino que se rigen por una cierta lógica e incluso aprenden de sus errores. Si, por ejemplo, un guerrero fracasa en su intento de llegar hasta Redhan desde un sitio determinado, la próxima vez, —si vive para contarlo— no repetirá la misma acción, sino que intentará otra cosa diferente. O si comprueba que todos sus intentos de ataque fracasan, se limitará a defender la zona que le ha sido asignada o, incluso, llegar a marcharse para pedir ayuda.

Más adelante, hablaremos detalladamente de los personajes y su forma de actuar y luchar, que es bastante compleja. Por ahora nos contentaremos con la exposición de la historia y de algunas de las ideas que salen en el juego. En el listado que aparece en la página anterior podemos ver una sencilla «demo» de la animación de uno de los personajes del juego, Gigántula, una simpática bestia peluda que lanza enormes bolas pegajosas y que da unos mordiscos de espanto.

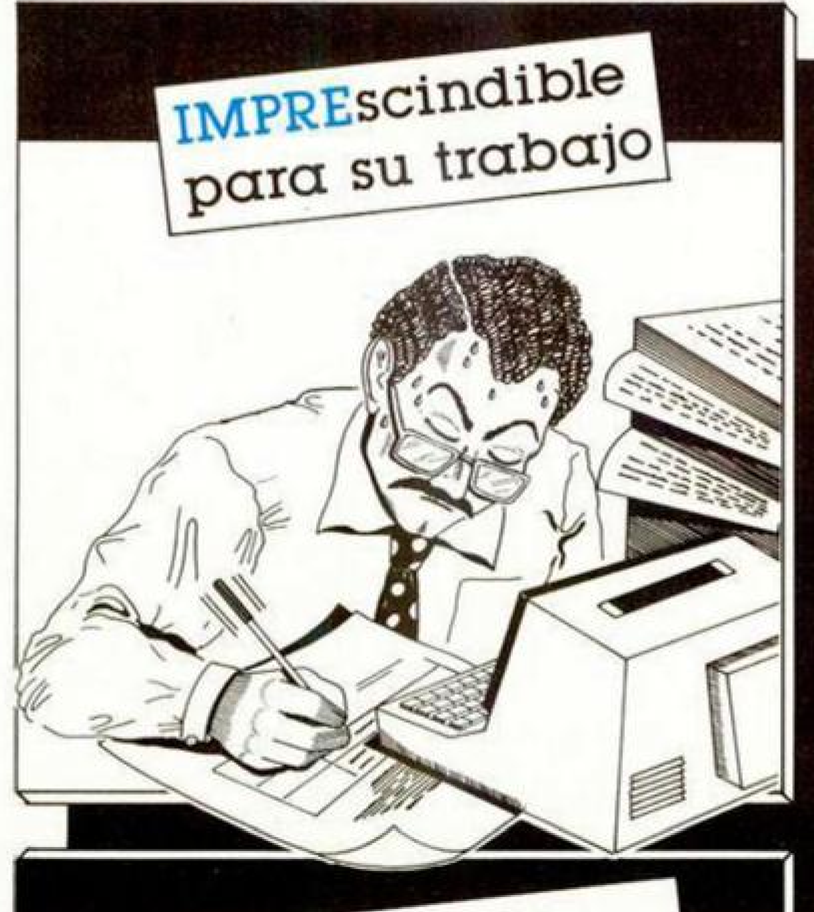
¡Hasta la próxima!

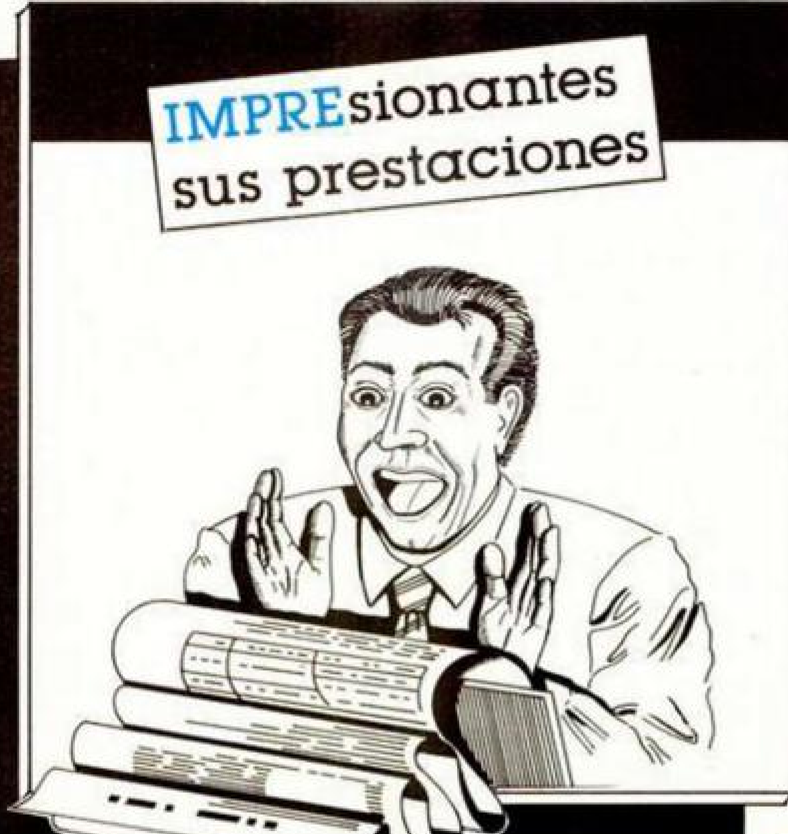
COMO UTILIZAR LA RUTINA

Para que os vayáis haciendo una idea de los magníficos gráficos que puede llegar a tener un programa comercial, a la vez que os percatáis del movimiento tan perfectamente conseguido, aquí tenéis una rutina, gentilmente cedida por los programadores de ERBE Software.

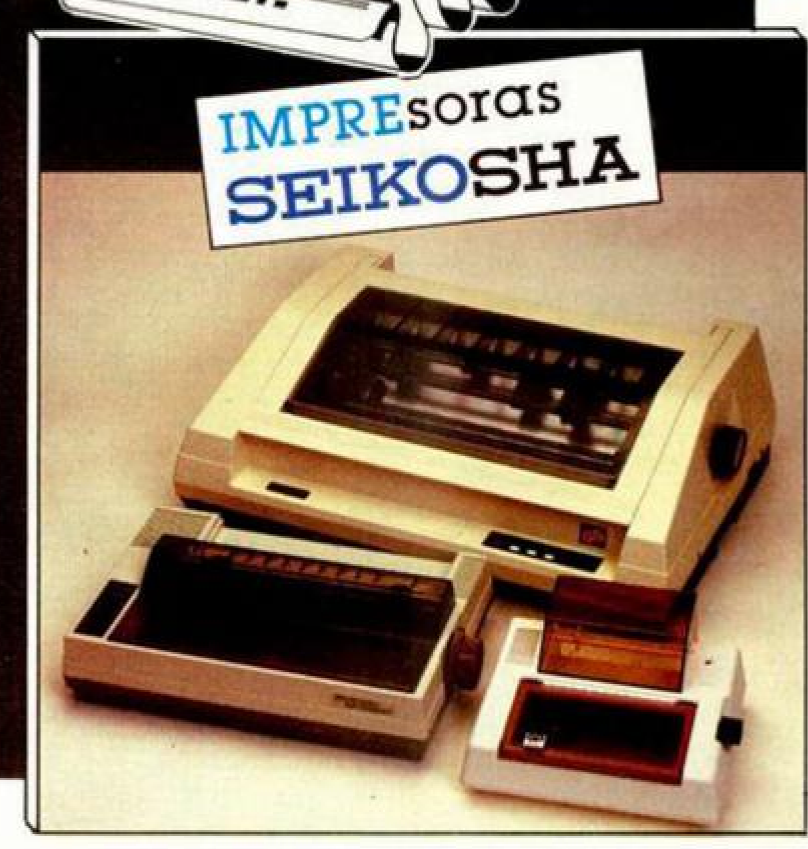
Para utilizarla habréis de seguir los siguientes pasos:

Teclear y salvar en cinta el programa 1, como SAVE «DEMO» LINE 1. A continuación, y utilizando el Cargador Universal de Código Máquina, teclear el listado 2, y realizar un DUMP en la dirección 30000. Salvar el Código Objeto generado indicando 30000 como dirección y 991 como número de bytes. Que lo disfrutéis.









trebol tel.(96)362 42 15

GP-50 • La pequeña 40 cps. Papel normal con interface paralelo, serial y Spectrum........ 17.990 ptas.
GP-700 * La de color 50 cps. 7 colores. 80 columnas. Tracción y fricción. Papel de 10 pulgadas .... 64.990 ptas.
SP-1.000 * La programable 100 cps.24 cps en alta calidad 96 cart. programables en RAM. Introductor hoja a hoja.♦........ 64.990 ptas.
SP-1.000AS La programable 100 cps.24 cps en alta calidad con interface RS-232. Introductor hoja a hoja.♦........ 59.900 ptas.
MP-1.300AI La polivalente 300 cps, 60 cps en alta calidad, interface paralelo y RS-232. Introductor hoja a hoja.♦&........ 119.900 ptas.
BP-5.200 * La de oficina 200 cps, 106 en alta calidad.Buffer 4K.Carro de 15".Tracción y fricción.♦........ 199.900 ptas.
BP-5.420 * La más rápida 420 cps. 106 cps en alta calidad. Buffer de 18K. Paralelo y RS-232.♦........ 339.900 ptas.

Interfaces: Serie RS-232C, Spectrum, IBM, COMMODORE, MSX, QL, Apple Macintosh, HP-IB
♦ Introductor automático de documentos opcional.
& Kit de color opcional.

* con interface paralelo
• con interface Spectrum

Nota: I.V.A. 12%, no incluido en los precios arriba indicados

Avda. Blasco Ibáñez, 116
Tel. (96) 372. 88. 89
Telex 62220 - 46022 VALENCIA

Muntaner, 60-2.º-4.ª
Tel. (93) 323. 32. 19
08011 BARCELONA

Agustín de Foxá, 25-3.º-A
Tels. (91) 733. 57. 00 - 733. 56. 50
28036 MADRID



LOS JUSTICIEROS DEL SOFTWARE

RASPUTIN

El espíritu del famoso Rasputín ha vuelto a revivir de la mano de Firebird, y esta semana les ha tocado a nuestros justicieros enarbolar la bandera de la ecuanimidad ante semejante personaje. Hemos de reconocer que la tarea no es nada fácil.

«Una idea original»

POSITIVO

Los gráficos son increíbles y la idea es muy original. Han conseguido un juego con el que te puedes pasar mucho tiempo divirtiéndote antes de llegar al final.

NEGATIVO

Es demasiado difícil, creo que el autor se ha pasado poniendo peligros en el camino del protagonista y le ha dotado de pocas armas. De todas formas, puede que sea precisamente esto lo interesante del juego.

PUNTUACION: 9

Pilar Arias

«Crea adicción»

POSITIVO

Los gráficos son muy buenos y posee bastantes pantallas. El color está bien tratado y se ven claramente todos los detalles. Es bastante difícil por lo que crea adicción.

NEGATIVO

Tal vez las instrucciones del juego no resulten lo suficientemente claras por lo que hay que descubrir cómo es su desarrollo. El sonido no resulta muy brillante.

PUNTUACION: 8

David de Llodio

«Buena presentación»

POSITIVO

Buena presentación. Sonido muy bueno a lo largo del juego, con una melodía al principio bastante aceptable. Cada vez que se mueve, nuestro personaje se cae o ataca. Los dibujos muy logrados al igual que las pantallas y el movimiento del personaje. Por la dificultad del mismo y la variedad de pantallas están garantizadas horas y horas de entretenimiento.

NEGATIVO

Se trata de un juego muy difícil de completar. Las instrucciones son poco explicativas.

PUNTUACION: 10

Fco. Javier Cano

«Entraña gran dificultad»

POSITIVO

Gráficos buenísimos. El movimiento es muy real y muy adictivo y la música del principio es bastante buena, al igual que cuando te matan.

NEGATIVO

Durante el juego no hay casi efectos sonoros. Es un juego bastante difícil y hace falta jugar mucho para pasar a otra dimensión.

PUNTUACION: 9

Roberto Carlos Alonso

«Gráficos bien elaborados»

POSITIVO

Los gráficos están bien elaborados. Es original por la estructura y la forma de desarrollarse el juego, pero su tema no lo es tanto. El sonido es aceptable aunque se podrían haber logrado mejores efectos sonoros. El movimiento es bueno, aunque algo lento.

NEGATIVO

Su enorme dificultad hace que disminuya la adicción y el interés por el juego. La visión de las pantallas resulta a veces bastante compleja.

PUNTUACION: 8

Ricardo Alonso

«Carece de sonido»

POSITIVO

Los gráficos son bastante buenos aunque se echa en falta la variedad de colores. El scroll está muy bien conseguido. El medidor de energía es muy original y la pantalla de presentación está muy bien realizada.

NEGATIVO

Carece de sonido. Las pantallas gráficamente son muy parecidas. La dificultad, junto con la lentitud, hacen que el juego sea uno más.

PUNTUACION: 6

José Galiana

«Movimiento complicado»

POSITIVO

Lo que cabe destacar es sin duda, los gráficos ya que además de presentar una gran variedad están muy bien conseguidos. También sobresale su alto nivel de adicción.

NEGATIVO

Es un programa que está en la línea del Alien-8, lo cual le quita originalidad al programa. Se podría haber mejorado el sonido y el movimiento que es algo complicado.

PUNTUACION: 8

Juan Markessinis

«Complejidad de pantallas»

POSITIVO

La idea es muy original al combinar la Edad Media con el futuro. Gráficos y movimientos muy bien realizados y ambientación muy acertada.

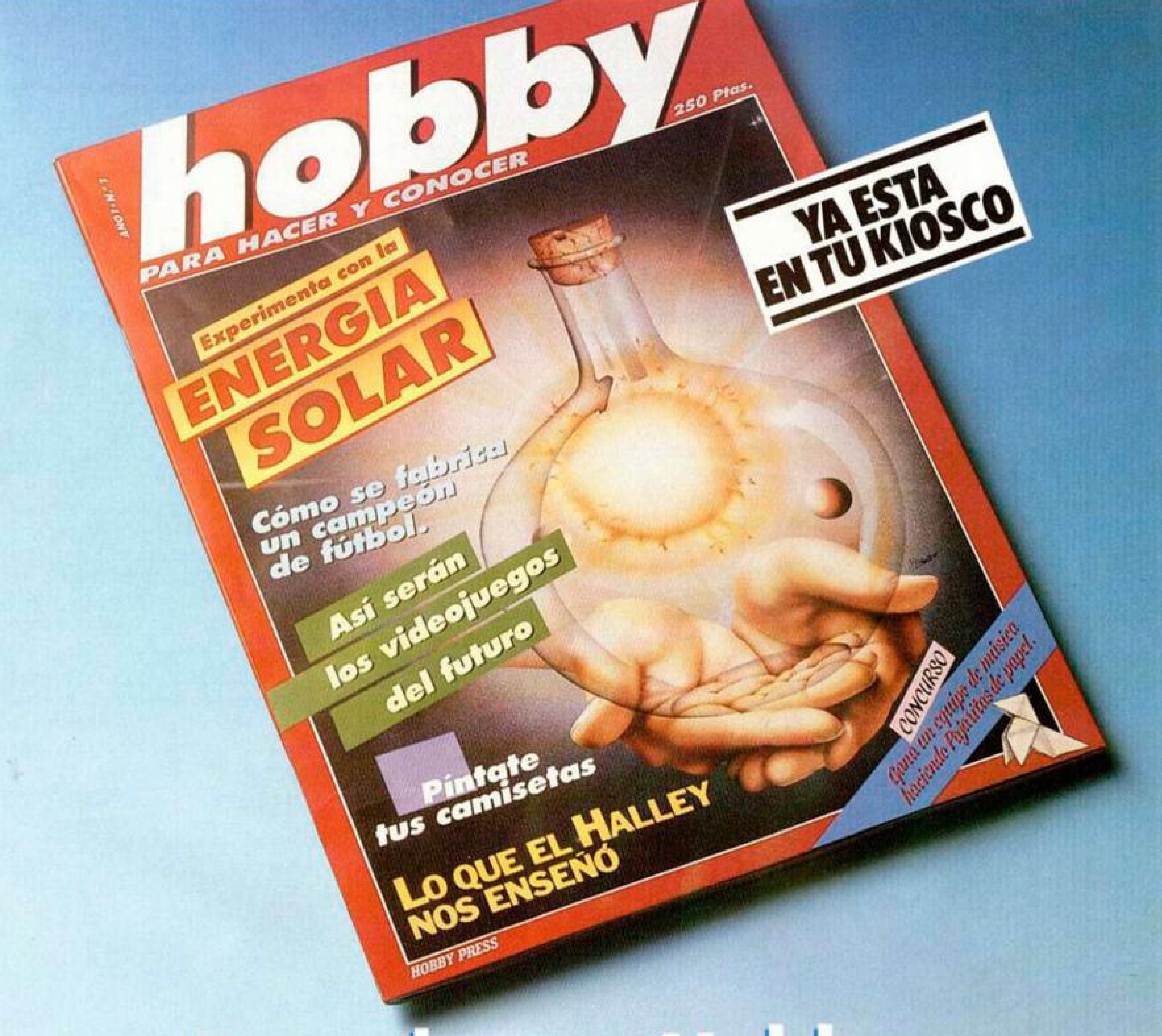
NEGATIVO

Habría que achacar al juego una excesiva complejidad en la mayoría de las pantallas que resultan prácticamente imposibles de completar.

PUNTUACION: 7

M.ª Antonia Gago





Descubre tu Hobby. La revista para hacer y conocer.

Ya ha llegado a tu kiosco la revista mensual que estabas esperando.

Por primera vez alguien pasa revista a tus aficiones más creativas.

Para saber lo que es bueno y conocer lo que es mejor.

Desde cómo adiestrar una paloma mensajera, cómo pintar tus camisetas de verano o experimentar con la energía solar.

Hay Hobby para todos los gustos y edades.

Todos los meses más de ochenta páginas a todo color, plenas de ideas, por sólo 250 ptas.

Haz algo por ti y por los que te rodean. Corre a tu kiosco y llévate Hobby.

HOBBY LO ENSEÑA TODO.

etc

HOBBY PRESS. Para gente inquieta.



NUEVO JOYSTICK SPEEDKING/KONIX

Sin duda se trata del joystick más suave y sensitivo que hemos tenido ocasión de manejar, a la vez que uno de los más precisos.

Sobre todo es importante indicar que se percibe claramente la sensación de que el movimiento ha sido ya realizado. De esta forma nos olvidamos del problema que se presenta con otros mandos en los cuales se tiende a presionar excesivamente, siempre con la duda de si el contacto se habrá realizado correctamente. En este caso, con un suave «click» se percibe el desplazamiento. Respecto a la construcción hay que destacar dos aspectos fundamentales.

En primer lugar, su diseño es totalmente ergonómico, lo que permite una perfecta adaptación al hueco de la mano, posibilitando el que éste sea firmemente asido y no se despegue de la mesa en el momento más inoportuno, impulsado por nuestros violentos movimientos. En este sentido es importante destacar que el Joystick se fabrica en dos modelos, uno para diestros y otro para zurdos. Internamente, su construcción es sumamente robusta, habiéndose dotado a los contactos de modernos microswitches (microinterruptores) que aseguran un período de uso de más de un millón de accionamientos para cada uno de ellos. En otras palabras, que podemos eliminar a la raza marciana a poco que nos descuidemos.

Por otra parte, podemos estar seguros de que no va a romperse fácilmente debido a que su manejo presenta una suavidad impresionante, siendo totalmente innecesario forzarlo, ya que responde a la menor insinuación.

Otra gran ventaja de este modelo, por no decir la más importante, es que resulta bastante barato en comparación con otros Joystick de similares prestaciones.

Este mando se encuentra disponible en España a través de SERMA, su distribuidor oficial, y costará unas 2.000 pts.



Aspecto que presenta el joystick para usuarios diestros.

Los microswitches interiores aseguran más de un millón de usos.



La perfecta adaptación del mando es total.

Protecciones aleatorias con el registro «R»

LA BIBLIA DEL «HACKER» (IX)

José Manuel LAZO

Siguiendo con el estudio detallado de los diferentes tipos de protecciones que se pueden llevar a cabo a nivel de código objeto, vamos a mostraros esta semana aquéllas que están directamente relacionadas con el registro de refresco o registro R.

El registro «R» es uno de los muchos que tiene el microprocesador de uso específico para él, en este caso para la memoria ya que se encarga de ir contabilizando la página de memoria que le toca ser refrescada por él mismo (para más información consultar los artículos de Primitivo en la sección Hardware).

Lo que a nosotros nos interesa es que su valor va variando secuencialmente con el tiempo, y muy rápido (relativamente), se puede decir que si consultamos su valor en un momento dado devuelve un número aleatorio, pero que para ciertas rutinas muy bien sincronizadas puede resultar predecible (vaya lío ¡eh!).

De esto se deduce que es muy sencillo que lo que se cargue de la cinta sea un montón informe de bytes y que, después de haberlos pasado por la piedra, oséase una rutina desenmascaradora, se conviertan en el verdadero código objeto limpio.

Una rutina desenmascaradora tiene un aspecto muy parecido a la del Cheksum salvo que todas las direcciones se Xorean con el registro «R» para producir el verdadero código objeto.

```
10         LD   HL,25000
20         LD   BC,40000
30         LD   A,0
40         LD   R,A
50 LOOP    LD   A,R
60         XOR  (HL)
70         LD   (HL),A
80         INC  HL
90         DEC  BC
100        LD   A,B
110        OR   C
120        JR   NZ,LOOP
130        RET
```

Rutina desenmascaradora.

El colmo del refinamiento viene cuando la rutina desenmascara otra que viene a continuación y pasa el control a la misma, la cual ya verdaderamente desenmascara el código limpio y entre ambas no hay ninguna inicialización del registro «R».

Por supuesto, ambas técnicas de protección se pueden mezclar y hasta incluso no hay nada que impida que esta última, en vez de producir el código a partir del registro «R» lo produzca a partir de la pantalla de presentación que acompaña al juego.

Esto podría ser así:

```
10         LD   HL,40000
20         LD   DE,16384
30         LD   BC,6912
40 LOOP    LD   A,(DE)
50         XOR  (HL)
60         LD   (HL),A
70         INC  HL
80         INC  DE
90         DEC  BC
100        LD   A,B
110        OR   C
120        JR   NZ,LOOP
130        RET
```

Un ejemplo

Seguro que ya estabais pensando que nos habíamos olvidado de explicar la manera en que están protegidos ciertos programas, pues no, y como el movimiento se demuestra andando aquí y ahora os vamos a exponer, como primicia mundial, la manera en que se protegió el EVERYONE'S A WALLY programa éste de MIKRO GEN.

De principio el programa se halla protegido con una rutina de carga de velocidad distinta a la normal, cuestión ésta que estudiaremos más adelante. La rutina cargadora se ubica en la última página de memoria y el bloque que se carga se solapa por encima del cargador con lo cual todo lo que se encuentre por encima de la cargadora no tiene sentido ya que es lo que se carga de la cinta.

Después de la carga se procede a un Cheksum de la memoria, incluyendo el cargador, para comprobar que no se ha tocado nada.

Luego se salta directamente a una rutinita ubicada en la memoria intermedia de impresora que se encarga de producir otra con unos valores situados después y otros ubicados en la pantalla de presentación mediante un sencillo, pero efectivo algoritmo.

Una vez que se ha producido esta rutina se pasa el control a la misma, la cual se encarga de desenmascarar todo el código que ha entrado de la cinta mediante el registro «R».

Llegados a este punto, ya se hace el salto al programa principal.

Vemos de esta forma cómo los programadores de MIKRO-GEN han imprimido en sus creaciones una serie de protecciones bastante completas y difíciles de desproteger. Además, hay que reconocer que la rutina de carga rápida que se utiliza para cargar el código está perfectamente hecha siendo, hasta incluso, más fiable en la carga que la de la ROM estándar. Esto es todo por esta semana...

MICRO Manía

Sólo para adictos

BRIAN BLOODAXE

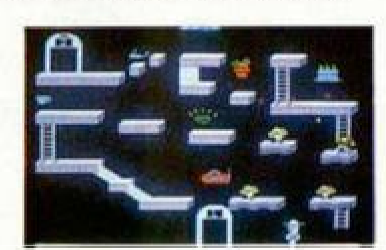
Este Brian Bloodaxe es un enloquecedor programa en el que los peligros y las amenazas se suceden continuamente con la intención de amargar la vida a este abnegado vikingo que pelea denodadamente en pos del trono de Bretaña.

Roberto Gómez Moreno nos envía, desde las mismísimas murallas de Avila, este interesante cargador que nos proporciona vidas no finitas. (Lo de infinitas está ya muy visto.)

```
10 CLEAR 64000
20 PRINT AT 9,2; "PLAY IN BLOODAXE TAPE"; AT 11,9; "FROM THE START"
30 RESTORE
40 FOR N=23296 TO 23374: READ A: POKE N,A: NEXT N
50 RANDOMIZE USR 23296
60 DATA 6, 3, 197, 221, 33, 0
70 DATA 0, 17, 0, 0, 62, 255
80 DATA 55, 205, 86, 5, 193, 16
90 DATA 239, 221, 33, 232, 254, 17
100 DATA 34, 1, 62, 255, 55, 205
110 DATA 86, 5, 62, 201, 50, 149
120 DATA 255, 58, 84, 92, 254, 92
130 DATA 40, 9, 33, 240, 92, 17
140 DATA 182, 92, 205, 229, 25, 205
150 DATA 3, 255, 49, 135, 144, 221
160 DATA 33, 24, 246, 17, 136, 19
170 DATA 62, 255, 55, 205, 86, 5
180 DATA 175, 50, 214, 103, 195, 224
190 DATA 96•
```

BOUNTY BOB STRIKES BACK

Como sabréis los asiduos a este adicto programa, cada vez que conseguimos pasar un cierto número de salas de esta endemoniada mina, aparece un cierto código. Angel Manuel Aranguren nos ha hecho llegar desde las calurosas costas de Benidorm las diferentes claves de acceso a dichas pantallas. Estas son las siguientes (por riguroso orden de aparición).



IHB	DVJ
LTO	OAQ
JDP	PHH
MLB	XNR

GUNFRIGHT

David Ogallar Feito, asiduo colaborador de esta maravillosa sección de esta no menos alucinógena revista, nos ha contado un pequeño truco para este sensacional programa de Ultimate. Este consiste en un poke con el cual podemos conseguir que los pistoleros no disparen en el duelo.

POKE 48121 , 33

La próxima semana..., más.

«AUTO-RUN» en Código Máquina

Algunas veces, al cargar un programa (PIMBALL), hay que dar LOAD ""CODE, pero al terminar de cargarse, se auto-ejecuta sin teclear ningún RANDOMIZE USR... ¿Cómo se consigue esto?

Rubén RODRIGUEZ - Guadalajara

☐ Efectivamente, existen algunos programas que se cargan como un solo bloque de C/M que se auto-ejecuta al terminar la carga, sin necesidad de ninguna orden desde el Basic. El sistema utilizado por estos programas consiste en empezar a cargar al principio de la RAM, es decir, a partir de 16384; con lo que cargan la pantalla, las variables del Sistema y el propio programa de una sola vez.

Para conseguir la auto-ejecución, se cargan determinados contenidos en determinadas variables del Sistema. Uno de los procedimientos más frecuentes es colocar en la variable «ERR-SP» la dirección de un byte de memoria que contenga, a su vez, la dirección de arranque del programa.

En el caso concreto del «PIMBALL», se utiliza un procedimiento algo más sofisticado. La variable «ERR-SP» contiene un «0» para que, si hay error de carga, se borre toda la memoria. El arranque se produce a través de la variable «CH-ADD» que debe contener la dirección del siguiente carácter a interpretar por el Basic. Esta variable se carga con el valor «23841». A partir de esta dirección, se encuentra una línea de Basic «ficticia» que contiene: RANDOMIZE USR 27392 que es la verdadera dirección de arranque del programa. De esta forma, cuando termina de cargarse, el Basic lee esta línea y la ejecuta, arrancando así el código máquina.

Sistema Operativo

¿Qué diferencia hay entre Sistema Operativo y Código Máquina?

¿Qué Sistema Operativo tiene el Spectrum?

El lenguaje Assembler que publican en el Curso de Código Máquina, ¿puede ser utilizado en otros ordenadores?

Juan - Pontevedra

☐ El código máquina es el conjunto de «unos» y «ceros» que entiende el microprocesador. Se podría decir que es el único lenguaje que éste entiende. Para los humanos, las instrucciones en este lenguaje nos resultarían imposibles de recordar, por ello, se utiliza el lenguaje Assembler que es una traducción del código máquina a una serie de instrucciones simbólicas.

Un Sistema Operativo es un conjunto de rutinas que sirven para que el usuario pueda manejar un ordenador. Normalmente, los Sistemas Operativos se escriben en Assembler; aunque recientemente, se están empezando a utilizar, también, lenguajes de alto nivel.

Existen Sistemas Operativos «standard», como por ejemplo, el CP/M que incorporan, entre otros, los ordenadores Amstrad. En el caso del Spectrum, el Sistema Operativo es el propio Editor de Basic que se encuentra implementado en ROM. Se trata de un Sistema particular y específico de Sinclair.

El lenguaje Assembler que publicamos en el Curso de Código Máquina, es el Assembler del Z-80, por lo tanto, vale para cualquier ordenador que utilice un microprocesador Z-80 como CPU; por ejemplo, todos los Amstrad. No obstante, para programar en Assembler no es necesario sólo saber el lenguaje, sino además, tener un gran conocimiento del ordenador sobre el que se va a hacer correr un programa.

Carga del GENS-3

¿Se carga de alguna manera especial en ensamblador GENS-3?

Alejandro GAETE - Madrid

☐ El Ensamblador GENS-3 puede cargarse en cualquier dirección de memoria. Llamemos «d» a la dirección donde queremos cargarlo. Primero hay que hacer: CLEAR (d-1). Luego se carga con: LOAD ""CODE d. Y se arranca, por primera vez, con: RANDOMIZE USR d.

Si se sale a Basic, se puede volver a entrar en el ensamblador con: RANDOMIZE USR (d + 61).

Nuestra portada

Tengo por costumbre sacar fotocopias de determinados artículos de la revista, para encarpetarlos separadamente, y así, si ando buscando «trucos», no tener que irme a la n.° 1 y ojearlas todas.

Por ello, me permito hacerles una sugerencia: Se trata de que, al iniciar una nueva serie de artículos (como ahora con rutinas de utilidad, Top Secret o Estrategia), lo digan al principio.

Francisco CASADO - Madrid

☐ Normalmente, hacemos esto que usted pide mediante la portada (es lo que se suele hacer en todas las revistas). En cada número, hay un tema que nosotros consideramos principal; a este tema le asignamos el dibujo de la portada que suele ser alusivo. Asimismo, se indica con unos titulares en un cuerpo más grande y con fondo de color.

Concretamente, respecto a los ejemplos que usted cita, en el número 42 la portada estaba dedicada al nuevo Curso de Código Máquina y a las fichas coleccionables. La portada del número 70 hacía referencia a la nueva serie de Estrategia. Finalmente, el número 71 tuvo su portada dedicada a «Top Secret», con biblia y todo.

No obstante, agradecemos su sugerencia y prometemos tenerla en cuenta.

«Nanodrive»

Tecleé el código fuente de vuestro magnífico programa «Nanodrive», mediante un ensamblador. He comprobado su buen funcionamiento en cualquier ubicación entre 32768 y 55600. Sin embargo, en una línea de Basic o como genuino código máquina ubicado sobre la Ramtop pero por debajo de la dirección de memoria menor antes citada, se produce una diferencia en los tiempos de envío al port FEh de los datos. Me gustaría saber dos cosas:

a) Por qué se produce esta diferencia en los tiempos de ejecución de una misma rutina según la zona de memoria donde se ubica.

b) Qué valores de las constantes temporizadoras habría que adoptar para que el programa funcionase correctamente en esa zona baja de memoria.

Carlos VILAR - Fuerteventura

☐ Además del Z-80, el Spectrum tiene otro dispositivo que accede a la memoria; se trata de la ULA. Esta se encarga de leer continuamente los archivos de presentación visual y enviarlos al televisor. Estos archivos se encuentran en los 16 primeros Ks. de

RAM. Por ello, la ULA tiene absoluta preferencia dentro de esta zona. Si el microprocesador quiere acceder a una zona de memoria donde está trabajando la ULA, ésta le para el reloj hasta que termina, con lo que resulta imposible calcular con precisión el tiempo de ejecución de una rutina si se la hace correr en los primeros 16Ks. de RAM.

Como norma general, todas las rutinas en código máquina que requieran una temporización exacta, deben correr en las direcciones de la ROM o en los 32Ks. superiores de la RAM. No hay forma de evitar esto ya que las paradas de reloj provocadas por la ULA dependen de tantos factores, que pueden considerarse aleatorias, tanto en su frecuencia como en su duración.

Programas de lectores

Estoy haciendo un programa de utilidades. En caso de que lo mande a su revista, ¿se publicaría en MICROHOBBY o en «Microhobby Cassette»?, ya que hace mucho tiempo que no veo publicar ningún programa de lector en su revista.

En caso afirmativo, ¿recibiría las 15.000 pesetas o no recibiría nada?

¿Cómo debería mandarlo?

Andrés F. HERNANDEZ - Murcia

□Los programas que se publican en la sección «Programas Microhobby» proceden de nuestros lectores. Por tanto, cada semana publicamos uno o dos programas escritos por algún lector.

Si su programa es seleccionado, por supuesto que recibirá las 15.000 ptas.

Puede remitirnos su programa grabado en una cinta de cassette y exento de protecciones. No es necesario que acompañe el listado, aunque sí conviene que nos adjunte un breve comentario sobre su funcionamiento.

Averías

Poseo un Spectrum 48K. Hace unos días lo puse y se veían, en la pantalla, puntitos y rayitas y no salía el mensaje de encendido. Lo desenchufé y lo volví a enchufar, pero seguía igual. ¿Qué puede tener?, ¿cuánto podría costar el arreglo?

Fernando C. PEREZ - Sevilla

□Con frecuencia, recibimos cartas de lectores que nos dicen que su ordenador se ha averiado y nos preguntan cuál puede ser la avería. Nada nos haría más felices que disponer de una «varita mágica» con la que tocar la carta y que el ordenador empezase a funcionar.

No obstante, en el número 69, publicamos un reportaje sobre los principales talleres donde reparan ordenadores Spectrum. Les recomendamos a todos los que se encuentren en este problema, que acudan o telefoneen a alguno de estos talleres, donde estamos seguros de que serán bien atendidos.



ATENCION

En el número 72 de Microhobby aparecían las bases para participar en el concurso-promoción de ERBE, en las que hay que hacer unas aclaraciones:

— El cupón que hay que recortar cada semana, deberá pegarse en su lugar correspondiente de la página 31 del n.º 72 que, una vez completada, deberá mandarse a nuestra nueva dirección (Hobby Press. Carretera de Irún kilómetro 12.400 28049 Madrid).

— Así mismo, hay que especificar que pueden enviarse dos logotipos de ERBE, dos de US Gold o uno de cada, indistintamente, también insertados en la misma página.

DE OCASION

● VENDO coche R/C eléctrico Tamiya con emisora 40 MHz., por 30.000 ptas., o bien lo cambio por monitor color o por un Spectrum 48 K con microdrive. Acepto otras ofertas. Escribir a Joaquín. Apartado de correos, 15149. 28080 Madrid.

● VENDO Zx Spectrum Plus, prácticamente sin estrenar por sólo 29.000 ptas. Interesados llamar al tel. 729 31 99 de Madrid. Preguntar por Pedro.

● URGE vender Zx Spectrum 16 K, con ampliación interna a 48 K, fuente de alimentación, cinta de Horizontes, cables y manual en castellano. También vendo Joystick Quick Shot IV con interface Indescomp programable, por sólo 34.000 ptas. Envío contra reembolso. Interesados escribir a Luis Jorge Pérez Rebolleda. C/ Corpus Christi, 41. 47005 Valladolid.



● VENDO Zx Spectrum 48 K con teclado Multifunción 1 (nuevo), cassette para ordenador, interface tipo Kempston, por sólo 44.000 ptas. Regalo libro de Basic y 3 revistas. Interesados llamar a Manuel Soler Llorca. Tel.: (965) 80 67 38. O bien escribir a la siguiente dirección: P. Santa María, 10. Villena (Alicante).

● VENDO T.C.R. 7205 con todos sus accesorios, rectas y curvas, un transformador, 4 coches y varios obstáculos, con dos mandos. Incluyo repuestos para los coches por sólo 10.000 ptas. También vendo una cámara fotográfica. Todo por sólo 15.000 ptas., lo vendo también por separado o bien lo cambio todo por un Spectrum o alguno de sus accesorios. Si te interesa llama al tel. 619 89 18 de Madrid. Preguntar por Pedro.

● VENDO/CAMBIO Quick Shot II más interface Kempston, lápiz óptico, bien lo cambio por un teclado multifunción Indescomp. Interesados llamar al teléfono (93) 241 73 76. Preguntar por Dani.

● TENGO las instrucciones de los siguientes programas: Avalon, Alien 8, Airwolf, Full Throttle, Olympicon, Atic At, etc. Desearía intercambiarlas por otras instrucciones como Babaliba, Saimazoom, Jet Set Willy, Sabre Wulf, etc. Al igual que mapas. Interesados pueden escribir a la siguiente dirección: Oriol Pitarch Bort. C/ Sanahuja, 53, Pta. 15. 8.°. 12001 Castellón de la Plana.

● VENDO Spectrum 48 K en buen estado, con sus respectivos libros, interface Joystick Quick Shot II, amplificador de sonido, cassette especial para ordenadores (tono y volumen) más de 50 revistas del Spectrum. Todo por el precio de 19.000 ptas. Interesados llamar al tel. (928) 25 94 61 o bien escribir a la siguiente dirección: Rafael Luis Benitez. Avda. Escaleritas, 41, A; 7.° C. 35011. Las Palmas de Gran Canaria.

● DESEARIA ponerme en contacto con todos los clubes de Spectrum que deseen ampliar su «plantilla» de socios. Interesados escribir a la siguiente dirección: Paco Estévez. Oleta Kalea, 1, 3.° B. Basauri (Vuacaya). O bien llamar al tel. (94) 440 43 86.

● VENDO monitor Zenith fósforo ámbar valorado en 30.000 ptas. Nuevo. Gastos de envío gratis. Por sólo 21.000 ptas. Alexis Gutiérrez. C/ Gutiérrez Rada, 2. Laredo (Cantabria). Tel.: (942) 60 62 25.

● VENDO Zx Spectrum 48 K con todos sus accesorios, cables, fuente de alimentación, manual en castellano más interface programable para joystick, Joystick Quick Shot III, más sintetizador de voz Currah. Todo por sólo 34.000.

● VENDO Spectrum 48 K, completo en buen estado, con 66 revistas MICROHOBBY. Regalo el Interface I. Interesados llamar al siguiente tel. (948) 24 53 38. Preguntar por Ana.

● VENDO vídeo-juegos Philips G-7000 con dos mandos, por 30.000 ptas. También quisiera contactar con usuarios del Spectrum. Interesados llamar al tel. 206 60 49 y preguntar por Eduardo. Madrid. VENDO Spectrum 48 K con todos los cables, interface, multijoystick, cassette especial para ordenador, Joystick Quick Shot II, cinta de demostración en español. Todo por 28.000 ptas. Incluyo manuales. Urge vender. Interesados llamar al tel. (91) 208 00 40. Preguntar por José Miguel.

● VENDO vídeo-juegos Intellivision. Precio a convenir. Interesados llamar al tel. (972) 20 82 23 de Gerona. Preguntar por Jordi.

● CAMBIO teclado original Zx Spectrum por los programas ensamblador Gens-3 y Mons 3 de Hisoft y Trans-Express o similar. Interesados pueden llamar al tel. (954) 70 45 26. Preguntar por Antonio de 20 a 22 horas.

● VENDO Interface joystick programable (Indescomp), nuevo, por 4.000 ptas. (negociables). Precio real 5.250 ptas. Permite controlar con joystick cualquier juego. Llamar al tel. 457 38 20 de Madrid. Preguntar por José.

● DESEARIA recibir una fotocopia de la tabla de códigos del juego Rocky. Pagaría unas 50 ptas. Interesados escribir a Jaime Dulcet Teixidió. San Jaime, 3, bajos. Ripollet (Barcelona). Tel. 692 24 86.

● VENDO Interface 1, más Microdrive para Zx Spectrum. Precio: 14.000 ptas. Interesados dirigirse a Alberto González Garre. Avda. Reyes Magos, 50, 4.° B. Alcalá de Henares (Madrid). Tel. 881 58 88.

● VENDO Sony Hit Bit 75-P 80 K Ram, con garantía, cables, manuales, reproductor de cassette por 50.000 ptas. Comprado en Dic-85. Escribir a Aitor Guisasola. Telletxe, 10, 3.° 1.ª. Algorta (Vizcaya). Tel. (94) 469 29 31.

● DESEARIA contactar con usuarios del Spectrum de 48 K y 128 K, para intercambio de ideas, trucos. Fernando Manuel Rodríguez Pardo. Renfe, 20, bajo izq. 04006 Almería.

● VENDO consola de vídeojuegos en perfecto estado por sólo 5.000 ptas. Interesados escribir a Juan Antonio M. de Francisco. Virgen del Templo, 22. San Fernando de Henares (Madrid).

● VENDO Zx Interface II, y lápiz óptico por sólo 15.000 ptas. todo. Adjunto libro de instrucciones. Interesados llamar al tel. 477 71 63 de Madrid. Preguntar por Manolo (mañanas).

● VENDO Spectrum 48 K en perfecto estado de funcionamiento con cables y transformador por 20.000 ptas. (negociables). Regalo cinta de Horizontes, manuales en castellano e inglés. Llamar a Daniel al tel. (93) 220 55 16. Barcelona.

● VENDO Zx 81 ampliado a 64 K, con cables, manual, etc. todo por sólo 17.000 ptas. Llamar tardes al tel. 856 58 50 de Madrid. Preguntar por Wassin.

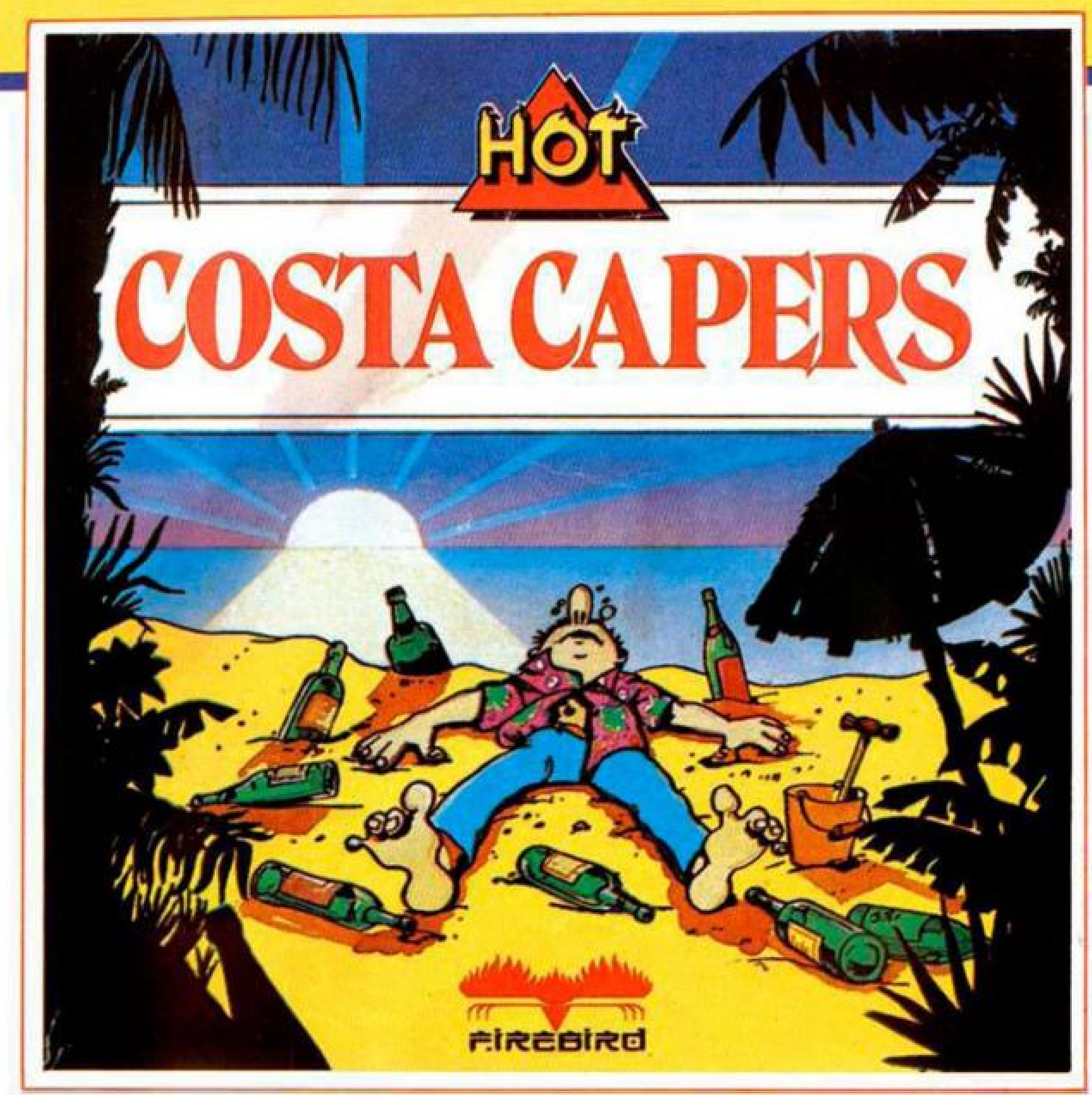
● COMPRO ordenador que esté en buen estado. También periféricos. Mandar precios. También monitor no importa el estado en el que se encuentre. Interesados escribir a la siguiente dirección: Faco. Javier Gallego. Calixto III, 1.°. Valencia. Tel. 326 71 77 (tardes).

● CONVIERTE tu Amstrad de 64 K de 320 K si tienes la expansión de 64 K DK'Tronics por sólo 14.000 ptas. Te la ampliamos a 256 K. También disponemos de expansiones de 256 K por sólo 27.000 ptas. Pide información a Octavio Bru Salas. Ribadavia, 10, bajo. 28029 Madrid.

● VENDO Spectrum 16 K sin usar por 12.000 ptas. Impresora Seikosha GP-50 por el precio de 16.000 ptas. Todo por 26.000 ptas. Escribir a José Fernando Toro. Monroy, 24, 4.° C. Madrid.

● VENDO cassette especial para ordenador por 4.000 ptas, teclado Multifunción Indescomp por 8.000 ptas., Impresora GP-50 S por 15.000 ptas. y un Spectrum por 23.000 ptas. de 48 K. Todo junto lo dejo en 45.000 ptas. y regalo algunas revistas. Interesados escribir a la siguiente dirección. Apartado de Correos 15149. 28080 Madrid.

● VENDO Spectrum Plus, seminuevo preparado para monitor con instrucciones de manejo y revistas. Todo por 34.000 ptas. Interesados contactar con Juan José Gómez Anaya. C/ San Macario, 20. 28021 Madrid.





CONDUCE A TED POR 57 PANTALLAS LLENAS DE AVENTURA Y EMOCION, MIENTRAS BEBE Y GASTA TODO SU DINERO, PERO NO DEJES QUE SE EMBORRACHE DEMASIADO O NO PODRA REVELAR LAS FOTOGRAFIAS DE SUS VACACIONES.

Próximamente en AMSTRAD

TAMBIEN LOS DEMAS JUEGOS DE LA SERIE









RECORTA Y ENVIA ESTE CUPON A: SERMA. C/ BRAVO MURILLO, N.º 377, 3.º A. 28020 MADRID

CANTIDAD: ____________________

NOMBRE Y APELLIDOS: ____________________

DIRECCION: ____________________

POBLACION: ____________________ PROVINCIA: ____________________

CODIGO POSTAL: ____________________ FORMA DE PAGO: ENVIO TALON BANCARIO ☐ CONTRA REEMBOLSO ☐